Universidade de Aveiro Departamento de Comunicação e Arte

Universidade do Porto Faculdade de Letras

Ano 2014

Elisabete Ferraz da Cunha

CONTRIBUTOS PARA A EFICÁCIA DO CLUSTERING USANDO O TAGGING SOCIAL

Universidade de Aveiro Departamento de Comunicação e Arte

Universidade do Porto Faculdade de Letras

Ano 2014

Elisabete Ferraz da Cunha

CONTRIBUTOS PARA A EFICÁCIA DO CLUSTERING USANDO O TAGGING SOCIAL

Tese apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Informação e Comunicação em Plataformas Digitais, realizada sob a orientação científica do Doutor Álvaro Reis Figueira, Professor Auxiliar do Departamento de Ciência dos Computadores da Faculdade de Ciências da Universidade do Porto e do Doutor Óscar Mealha, Professor Associado do Departamento de Comunicação e Arte da Universidade de Aveiro.

Apoio financeiro da FCT e do FSE no âmbito do III Quadro Comunitário de Apoio.

o júri

presidente

Doutor **Carlos Alberto Diogo Soares Borrego**, Professor Catedrático da Universidade de Aveiro.

Doutora **Cândida Fernanda Antunes Ribeiro**, Professora Catedrática da Faculdade de Letras da Universidade do Porto.

Doutor **Óscar Emanuel Chaves Mealha**, Professor Associado com Agregação da Universidade de Aveiro. **(Orientador)**.

Doutor **Paulo Jorge de Sousa Gomes**, Professor Auxiliar da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Doutor **Álvaro Pedro de Barros Borges Reis Figueira**, Professor Auxiliar da Faculdade de Ciências da Universidade do Porto. **(Coorientador)**.

Doutora **Ana Alice Rodrigues Pereira Baptista**, Professora Auxiliar da Escola de Engenharia da Universidade do Minho.

Doutor **Joaquim Manuel Henriques de Sousa Pinto**, Professor Auxiliar da Universidade de Aveiro.

Doutora **Isabel de Fátima Silva Azevedo**, Professora Adjunta do Instituto Superior de Engenharia do Porto.

agradecimentos

Ao Professor Doutor Álvaro Reis Figueira, obrigada, pelos contributos, pela disponibilidade, por ter permanecido nesta longa jornada ao meu lado, por não ter deixado de acreditar.
Ao Professor Doutor Óscar Mealha, obrigada pelos contributos e disponibilidade.
Ao José Devezas, agradeço a partilha de conhecimentos que em muito ajudou a definir o caminho trilhado nesta tese.
Ao Jorge Silva e colegas do *Breadcrumbs*, Henrique, Mário, José Carlos, e Nuno, obrigada pela constante disponibilidade e ajuda na implementação dos algoritmos.
Aos meus colegas da Escola Superior de Educação que, com amizade me apoiaram durante todo este processo.
Agradeço ainda à Fundação para a Ciência e Tecnologia pelo financiamento concedido a esta investigação.
Ao Pedro, ao Pedrinho e restante família, que com amor e motivação tornaram isto possível.

palavras-chave Clustering; tagging social; eficácia; distância semântica; k-means, k-C.

resumo

Nos últimos anos temos vindo a assistir a uma mudança na forma como a informação é disponibilizada online. O surgimento da web para todos possibilitou a fácil edição, disponibilização e partilha da informação gerando um considerável aumento da mesma. Rapidamente surgiram sistemas que permitem a coleção e partilha dessa informação, que para além de possibilitarem a coleção dos recursos também permitem que os utilizadores a descrevam utilizando *tags* ou comentários. A organização automática dessa informação é um dos maiores desafios no contexto da web atual. Apesar de existirem vários algoritmos de *clustering*, o compromisso entre a eficácia (formação de grupos que fazem sentido) e a eficiência (execução em tempo aceitável) é difícil de encontrar.

Neste sentido, esta investigação tem por problemática aferir se um sistema de agrupamento automático de documentos, melhora a sua eficácia quando se integra um sistema de classificação social.

Analisámos e discutimos dois métodos baseados no algoritmo *k-means* para o *clustering* de documentos e que possibilitam a integração do *tagging* social nesse processo. O primeiro permite a integração das tags diretamente no *Vector Space Model* e o segundo propõe a integração das *tags* para a seleção das sementes iniciais. O primeiro método permite que as *tags* sejam pesadas em função da sua ocorrência no documento através do parâmetro *Social Slider*. Este método foi criado tendo por base um modelo de predição que sugere que, quando se utiliza a similaridade dos cossenos, documentos que partilham *tags* ficam mais próximos enquanto que, no caso de não partilharem, ficam mais distantes. O segundo método deu origem a um algoritmo que denominamos k-C. Este para além de permitir a seleção inicial das sementes através de uma rede de *tags* também altera a forma como os novos centróides em cada iteração são calculados. A alteração ao cálculo dos centróides teve em consideração uma reflexão sobre a utilização da distância euclidiana e similaridade dos cossenos no algoritmo de *clustering k-means*.

No contexto da avaliação dos algoritmos foram propostos dois algoritmos, o algoritmo da "*Ground truth* automática" e o algoritmo MCI. O primeiro permite a deteção da estrutura dos dados, caso seja desconhecida, e o segundo é uma medida de avaliação interna baseada na similaridade dos cossenos entre o documento mais próximo de cada documento.

A análise de resultados preliminares sugere que a utilização do primeiro método de integração das tags no VSM tem mais impacto no algoritmo k-means do que no algoritmo k-C. Além disso, os resultados obtidos evidenciam que não existe correlação entre a escolha do parâmetro SS e a qualidade dos clusters. Neste sentido, os restantes testes foram conduzidos utilizando apenas o algoritmo k-C (sem integração de tags no VSM), sendo que os resultados obtidos indicam que a utilização deste algoritmo tende a gerar clusters mais eficazes.

keywords Clustering; social tagging; effectiveness; semantic distance; k-means, k-C.

abstract

In recent years there has been a change in the way information is displayed online. The generalized access to the world wide web allowed an easy production, editing, distribution and sharing of the information, resulting in a massive increase of data. Thereafter were created systems thought to collect and share that information, as well as allowing the users to tag or comment the data. The automatic organization of that information is one of the biggest challenges in the current Web context. Despite the existence of several clustering algorithms, the commitment between effectiveness (forming groups that make sense) and efficiency (doing so in an acceptable running time) is difficult to achieve.

Therefore, this investigation intends to assess if a document clustering system improves it's effectiveness when integrating a social classification system.

We have analyzed and discussed two methods for clustering documents, based on the k-means algorithm, which allows the integration of social tagging in the clustering process. The first method allows integrating tags directly into the Vector Space Model and the second proposes the integration of tags to select the initial seeds. The first method allows tags to be weighted according to their occurrence in the respective document through the Social Slider parameter. This method was based on a predicting model which states that when using cosine similarity, the documents sharing tags are closer and when not sharing tags, documents are more distant. The second method generated an algorithm named k-C. In addition to allowing initial seed selection through a network of tags, it also changes the way new centroids are calculated in each iteration. The change in centroid calculation came from the use of Euclidian distance and cosine similarity in the k-means clustering algorithm.

Considering algorithm creation and assessment, two algorithms were proposed: the "Automatic Ground Truth" algorithm and the "MCI" algorithm. The first one allows detecting the data structure, if unknown; and the second one is an internal evaluation measure based on cosine similarity between the document closest to each document.

The analyses of the preliminary results suggests that using the first tag integration algorithm method on the VSM has a bigger impact on the k-means algorithm than on the k-C algorithm. Besides, the obtained results show that there is no correlation between the SS parameter choice and the quality of the clusters. In this sense, the tests were made using only the k-C algorithm (without tag integration on the VSM) and the results indicated that using this algorithm results in the creation of more effective clusters.

Sumário

Índice de Figuras

Índice de Tabelas

Índice de Gráficos

Índice de Esquemas

Introdução

Atualmente é reconhecido que o desenvolvimento da *World Wide Web* simplificou a disponibilização de conteúdos, gerando um considerável aumento da informação.

Para tirar o máximo partido da sociedade da informação, não é suficiente saber que a informação cresce a um ritmo exponencial. É, acima de tudo, essencial pensar que neste novo paradigma tecnológico, informação mais informação gera conhecimento. Como refere Castells (2000), a atual revolução tecnológica não é caracterizada pela centralidade do conhecimento e da informação, mas sim pela forma como esse conhecimento é utilizado na produção de novos conhecimentos e dispositivos que permitam novas formas de processar e comunicar essa informação.

Já Vannevar Bush no artigo "*As We May Think*" (Bush, 1945), apela para a necessidade de tornar o conhecimento humano colecionável e mais acessível. Este afirma que o carácter científico de classificação de informação utilizado nos sistemas de armazenamento de dados é bastante distante dos processos utilizados pela mente humana. Assim, em oposição a métodos de classificação numérica e/ou alfabética, defende que a mente humana efetua as suas pesquisas através de associações, percorrendo "os intrincados caminhos criados pelo cérebro" (capítulo 6, para. 2) na rede neuronal. É neste sentido que propõe o desenvolvimento do Memex, um mecanismo idealizado para auxiliar a memória humana que permitiria ao utilizador guardar e organizar mecanicamente "todos os seus livros, discos e comunicações para que os possa consultar de forma rápida e flexível" (capítulo 6, para. 4). No fundo, tratar-se-ia de um "suplemento alargado de memória"**.**

Com o aumento exponencial da produção de informação, levantam-se questões associadas à organização da mesma. Criar as condições para que novo conhecimento seja gerado está diretamente dependente da identificação de padrões que permitam relacionar a informação criando grupos de informação relacionada. Neste sentido, o *Text Mining*, campo interdisciplinar baseado em *Information Retrieval*, *Machine Learning*, *Data Mining*, Estatística, Linguística Computacional e *Natural Language Processing* (NLP), deteta padrões em dados não estruturados e compreende, entre outras técnicas, o *Clustering* de texto. Este consiste num processo não supervisionado que permite organizar documentos em grupos denominados de "*clusters*". Contudo, e apesar de existirem atualmente um grande número de algoritmos de *clustering*, os principais problemas relacionam-se com a eficácia dos agrupamentos, isto é com a qualidade dos grupos e escalabilidade dos algoritmos, se mantêm eficiência aceitável para um determinado input.

A emergência de importantes interações on-line abre também espaço à análise sobre a forma como os utilizadores podem contribuir para uma melhoria no sistema de agrupamento automático. Recorrentemente, na *Web*, são classificados vídeos, associadas palavras-chave a imagens, artigos, blogues, *bookmarks*, *Urls*, e desta forma são fornecidas descrições que permitem uma indexação através da linguagem natural. Surge assim o *tagging*, ou seja, a anotação de Recursos pelos utilizadores utilizando *tags*, definidas como textos arbitrários que descrevem os documentos. O *tagging* Social é o *tagging* em ambiente online onde as *tags* usadas pelos utilizadores estão disponíveis para os restantes utilizadores (Lohmann, 2011). Este, visto como um signo de comunicação, tem assumido um papel de relevo na organização da informação através da cooperação.

É neste sentido que a integração da classificação social no processo de agrupamento automático surge como uma hipótese a testar.

Do ponto de vista deste programa doutoral, este trabalho insere-se no âmbito da área científica de Ciências da Informação. Segundo Borko (1968) pode ser definida como:

> A disciplina que investiga as propriedades e o comportamento da informação, as forças que regem o fluxo informacional tendo em vista acessibilidade e usabilidade ótimas. Está relacionada com um corpo de conhecimento que abrange a origem, coleta, organização, armazenamento, recuperação, interpretação, transmissão, transformação e utilização da informação. Isto inclui a investigação das representações da informação quer nos sistemas naturais como

> artificiais, o uso do código para uma transmissão eficiente das mensagens e o estudo dos instrumentos e técnicas de processamento de informação como os computadores e os seus sistemas de processamento. É uma ciência interdisciplinar que deriva e está relacionada com disciplinas como a matemática, lógica, a linguística, a psicologia, ciência dos computadores, pesquisa operacional, artes gráficas, comunicação, biblioteconomia, gestão entre outras. Contempla uma componente de ciência pura que questiona sem preocupação com a aplicação e uma componente de ciência aplicada que desenvolve serviços e produtos (p. 3).

Neste sentido, esta investigação pretende apresentar contributos no contexto da organização da informação de forma não supervisionada, ao mesmo tempo que cria condições que facilitam a recuperação da mesma. Por outro lado, do ponto de vista da própria investigação, também pretendemos apresentar contributos no contexto da avaliação, uma vez que a metodologia a ser seguida parte da análise da natureza do tagging social como signo de comunicação online e portanto terá como base as Ciências e Tecnologias da Comunicação.

1.1. Complexidade Espácio-Temporal

O desenvolvimento de algoritmos de *clustering* eficazes, isto é que satisfaçam os interesses de um utilizador ou de uma comunidade de utilizadores, requer também que sejam identificados quais os recursos necessários para executar um algoritmo, tanto do ponto de vista da complexidade espacial como da complexidade temporal. A primeira diz respeito ao espaço em memória necessário, a segunda ao tempo de execução, dependendo da quantidade de dados usados como input. Portanto, é necessário averiguar se os algoritmos são eficientes (cumprem em tempo útil) e escaláveis (mantêm eficiência/eficácia aceitável) para um determinado número de documentos.

A título de exemplo, quando se compara a complexidade de dois algoritmos usando a medida de tempo – complexidade temporal – pretendemos averiguar qual é o mais rápido. Apesar da velocidade estar dependente da máquina e da linguagem de programação utilizada, vamos abstrair-nos destes factos e analisar apenas o número de operações necessárias para executar o algoritmo para um determinado *input*.

As ordens de complexidade mais comuns são:

- 1 – tempo de execução constante
- $Logn$ – tempo de execução logarítmico
- n – tempo de execução linear

- n^2 - tempo de execução quadrático
- n^3 - tempo de execução cúbico
- 2^n - tempo de execução exponencial (ineficientes)

Prever com rigor o tempo de execução de um algoritmo é uma tarefa difícil. Usualmente identificam-se as operações dominantes e exprime-se o resultado usando a notação do "Grande O".

Definição: $T(n) = O(f(n))$ e lê-se: $T(n)$ é de ordem $f(n)$ se e só se existem constantes positivas c e n_0 tal que $T(n) \leq c \times f(n)\ \forall\ n > n_0$.

De um modo informal, podemos dizer que o símbolo O seleciona o termo que mais contribui para o valor que uma expressão pode tomar, ignorando os fatores constantes desse termo. Por exemplo se $f(n) = 2n^2 + 3n + 5$, então $O\big(f(n)\big) = O(n^2)$.

Tabela 1: Tempo de execução da função custo de uma máquina que executa 10^9 passos por segundo ($\sim 1\ GHz$) (tabela obtida online[1] p. 10).

Tamanho	log_{2^n}	n	$nlog_{2^n}$	n^2	n^3	2^n
10	3,322 ns	10 ns	33 ns	100 ns	1 μs	1 μs
20	4,322 ns	20 ns	86 ns	400 ns	8 μs	1 ms
30	4,907 ns	30 ns	147 ns	900 ns	27 μs	1 s
40	5,322 ns	40 ns	213 ns	2 μs	64 μs	18,3 min
50	5,644 ns	50 ns	282 ns	3 μs	125 μs	13 $dias$
100	6,644 ns	100 ns	664 ns	10 μs	1 ms	$40{,}1^{12}$ anos
1000	10 ns	1 μs	10 μs	1 ms	1 s	
10000	13 ns	10 μs	133 μs	100 ms	16,7 min	
100000	17 ns	100 μs	2 ms	10 s	11,6 $dias$	
1000000	20 ns	1 ms	20 ms	16,7 min	31,7 $anos$	

As funções polinomiais crescem muito mais devagar do que as funções exponenciais e fatoriais. Considera-se que um algoritmo é eficiente se a sua complexidade temporal for $O\big(n^k\big), \forall\ k \in \mathbb{Z}$, e é considerado ineficiente se a sua complexidade cresce mais rapidamente que $n^k\ \forall\ k \in \mathbb{Z}$.

Na Tabela 1 podemos ver o tempo de execução em função da função custo de uma máquina que executa 10^9 passos por segundo ($\sim 1\ GHz$). Por exemplo, para 2^n se o

[1] http://chiusella.polito.it/www.testgroup.polito.it/images/teaching/02LTKJA/1-2_6x.pdf

tamanho do input for 50 já são necessários 13 dias para executar o algoritmo, sendo que para n=100 é considerado intratável.

1.2. Folksonomias

O termo “folksonomia” foi criado por Thomas Vander Wal (2007) e deriva da aglutinação dos termos *folk* (povo) e *taxonomy* (taxonomia). Subjacente fica a ideia da criação de uma taxonomia pelas pessoas, mas taxonomias dependem de uma hierarquia estável definida anteriormente (DYE, 2006), o que evidentemente não acontece numa *folksonomia*, onde a associação de *tags* não está dependente de uma hierarquia.

No contexto da inteligência coletiva de Pierre Lévy (1997), a ideia de que todos podem contribuir para a construção do conhecimento é de suma importância quando perspetivamos a possibilidade de integrar num sistema automático a classificação social. A intervenção do sentido crítico de cada um, poderá no todo, criar dinâmicas que ajudem a organizar a informação e torná-la mais relacionada, sobretudo quando a dimensão do "todo" pode corresponder a centenas de milhares, milhões, ... de documentos.

Pretendemos criar algoritmos de *clustering*, que incluam a informação disponibilizada pelos utilizadores através do tagging Social, no sentido de averiguar o impacto na eficácia dos agrupamentos produzidos.

Assim, esta investigação pretende contribuir para uma compreensão mais profunda sobre a influência da utilização da classificação social como forma de auxiliar o processo de classificação automática e de que forma a natureza das *tags* nos permite determinar a eficácia desses agrupamentos.

1.3. Problemática de Investigação

O trabalho de investigação proposto visa a modelação de um sistema de agrupamento automático de documentos de acordo com a sua proximidade semântica. Este sistema deverá operar em conjunto com uma base de dados de documentos digitais. O seu funcionamento apoia-se na extração e análise do texto presente nos documentos, sendo todo o processo de agrupamento realizado automaticamente, isto é, de forma não supervisionada.

A este sistema será feita a integração de uma classificação social, na qual se permite que outros utilizadores registem, facultativamente, termos a eles associados (*tags*).

Questão de investigação: Um sistema misto de classificação (integração de classificação automática e social) é mais eficaz do que um sistema que integre somente o agrupamento automático de documentos?

Pretende-se durante o programa de trabalhos modelar matematicamente o sistema, percebendo as partes que nele são escaláveis e entender a que nível o são; propor novos algoritmos de agrupamento e descrever a sua associada complexidade espácio-temporal.

1.4. Finalidades e Objetivos Esperados

O objetivo geral do presente programa de trabalhos passa pela modelação matemática de um sistema capaz de fazer o agrupamento automático de documentos utilizando a informação providenciada pelos utilizadores através do *tagging* social. Este objetivo geral integra 4 objetivos parcelares e específicos:

- Estudar e analisar algoritmos para realizar o *clustering* de documentos de forma eficiente e escalável;
- Estudar processos para redução de dimensão espacial e para pré-tratamento de dados;
- Estudar e criar uma possível integração de uma classificação social, com o agrupamento automático de documentos, baseada no *tagging* social;
- Verificar a eficácia e escalabilidade do objetivo anterior;

1.5. Propósito e Método de Investigação

No que diz respeito ao propósito, esta investigação classifica-se como básica (ou fundamental) que segundo Carmo e Ferreira (2008) tem por finalidade estabelecer princípios gerais a partir do desenvolvimento de uma teoria. Assim, através da pesquisa experimental, visa-se a otimização de algoritmos que tornem a classificação automática e social de documentos num procedimento eficaz.

Quanto ao método, esta investigação é considerada do tipo experimental que é, segundo Carmo e Ferreira (2008), "descrito como aquele que é conduzido para rejeitar ou aceitar hipóteses relativas a relações causa-efeito entre variáveis" (p. 243). Esta escolha é ainda fundamentada por Feldman e Sanger, que defendem que, em consequência do problema da categorização de texto não estar bem definido, resulta que "a performance de classificadores de texto pode ser avaliada apenas experimentalmente" (Feldman & Sanger, 2007).

1.6. Organização Geral

Este trabalho encontra-se estruturado em 6 capítulos. Depois desta Introdução, segue-se a revisão de literatura sobre algoritmos de *clustering*.

No capítulo 2 fazemos a revisão de literatura sobre o *tagging* social e enquadramos a natureza das *tags* na perspetiva da Teoria Semiótica.

No capítulo 3 propomos dois métodos para integrar as *tags*. O primeiro método integra as *tags* no vetor dos documentos através de um parâmetro chamado Social *Slider* (SS) que possibilita que seja dada importância diferenciada às *tags* em função da sua ocorrência no documento. Apresenta-se ainda o modelo teórico de previsão do impacto da integração das *tags*. O segundo modelo de integração é baseado na deteção de comunidades numa rede de *tags*, permitindo uma escolha cuidada das sementes. Para além disso, é feita a análise de complexidade do algoritmo proposto.

No capítulo 4 descrevemos as opções e procedimentos de caráter metodológico. Avaliamos os algoritmos quanto à sua eficácia e fazemos uma reflexão sobre possíveis reformulações dos algoritmos propostos, quer para melhorar a sua eficácia quer a sua eficiência.

Por fim, o capítulo 5 é constituído por conclusões finais onde são resumidos os principais parâmetros e conclusões desta dissertação.

Capítulo 1
Clustering

A classificação automática de texto não é um problema recente. Desde há várias décadas foi adotada por uma importante área de investigação chamada *Natural Language Processing* (NLP). Esta desenvolveu várias técnicas inspiradas na linguística em que o texto é desmontado sintaticamente, usando informação gramatical e lexical, permitindo que posteriormente a informação daí resultante seja interpretada semanticamente. O NLP pode ser profundo (partindo cada uma das partes de todas as frases e tentando agrupá-las semanticamente) ou superficial (partindo apenas algumas passagens e frases ou produzindo apenas análises semânticas limitadas), podendo ainda usar meios estatísticos para desambiguar o sentido das palavras. Tende a focar-se num documento (ou pedaço de texto) de cada vez, consumindo muitos recursos computacionais (Kao & Poteet, 2005). Os resultados dos primeiros trabalhos de pesquisa surgiram na década de 50, estando entre eles o artigo publicado por Alan Turing, intitulado "*Computing Machinery and Intelligence*" (Turing, 1950). Durante a década de 60 assistiu-se ao aparecimento de um número significativo de sistemas para compreender a linguagem natural, entre eles está o SHRDLU criado por Terry Winograd no MIT.

Segundo Feldman e Sanger (2007), o *Text Mining* surge mais recentemente e é um campo interdisciplinar que se baseia em *Information Retrieval*, *Machine Learning*, *Data Mining*, Estatística, Linguística Computacional e *Natural Language Processing* (NLP). Pode ser definido como um processo de conhecimento intensivo no qual o utilizador interage com uma coleção de documentos ao longo do tempo, usando um conjunto de ferramentas de análise. De maneira análoga ao *Data Mining*, o *Text Mining* visa extrair

informações úteis a partir de fontes de dados, identificando e explorando padrões. Contudo, diferentemente do *Data Mining,* a deteção de padrões também é feita em dados não estruturados, ou seja, ao longo de todo o texto.

Assim, o *Text Mining* compreende (Feldman & Sanger, 2007):

- Categorização de texto de acordo com um conjunto fixo de categorias onde, através de alguns exemplos de treino, o objetivo do sistema está em aprender a classificar automaticamente novos documentos;
- *Clustering* de texto que consiste num processo não supervisionado através do qual documentos são organizados em grupos denominados *clusters* - pois pretende-se que sejam detetados padrões permitindo que documentos similares sejam agrupados no mesmo *cluster*;
- Sistemas de extração da informação que assumem cada vez mais importância sobretudo quando é impossível ao utilizador ler toda a informação disponível;
- *Automatic Summarization* consistindo na criação de uma versão mais curta do texto através de um programa computacional.

Esta área ganhou especial relevância a partir dos anos 90 em consequência do armazenamento digital da internet e mais recentemente com o surgimento da *Web* 2.0 que provocou uma crise de excesso da informação. Com o desenvolvimento da *Web* 2.0 foram criadas as condições para que todos publiquem as suas ideias online, o que gerou a necessidade de descrever e organizar uma enorme quantidade de informação. Assim, tornou-se premente a procura de algoritmos mais rápidos para permitir que a recuperação da informação seja feita de forma eficaz e eficiente.

Neste sentido, neste capítulo aprofundaremos o estudo do *clustering*, começando por definir o problema geral de *clustering*. Segue-se uma breve reflexão sobre os conceitos documento e informação. Descrevem-se de seguida os vários métodos para representar documentos de texto bem como as medidas de similaridade frequentemente utilizadas para implementar os algoritmos de *clustering*. Por fim, são descritos diferentes tipos de algoritmos de *clustering*, apontando potencialidades e limitações relevantes.

1.1. Problema Geral de *Clustering*

Os problemas de *clustering* são na sua essência problemas de otimização. Espera-se que um bom *clustering* consiga colocar juntos documentos similares e simultaneamente separar os que não são similares (Feldman & Sanger, 2007).

1.2. Documento e Informação

Segundo Feldman e Sanger (2007), sob um ponto de vista prático, um documento, contendo exclusivamente informação de natureza textual, pode ser informalmente definido como uma unidade de informação textual dentro de uma coleção que normalmente, ainda que não necessariamente, se correlaciona com documentos do "mundo real" como por exemplo trabalhos de pesquisa, memorandos legais, e-mails, artigos manuscritos, comunicados de imprensa ou notícias.

Segundo Silva (2000) “O empirismo dominante e o excesso de senso comum têm tornado inextricável documentação e informação, não permitindo a necessária e conveniente distinção dos conceitos em jogo” (p. 3), sendo por isso frequente que se tomem os dois conceitos como sinónimos, quando segundo Silva (2006), documento corresponde à materialização da informação que potencia a comunicação, por ser considerado um estado intermédio de criação da informação até atingir o processo de comunicação.

Assim, no contexto desta investigação, o documento de texto surge como a unidade sobre a qual se exige uma representação de modo a permitir a sua organização automática, para posterior recuperação de acordo com as necessidades do utilizador.

1.3. Representação de Documentos de Texto

O modelo a adotar para a representação de documentos de texto é baseado num modelo de proximidade espacial (*Vector Space Model*), considerando-se cada documento como um vetor desse espaço (Salton, Wong, & Yang, 1975). A representação dos documentos em vetores, surge como uma necessidade de processar a informação, já que seria impraticável fazê-lo no seu formato original.

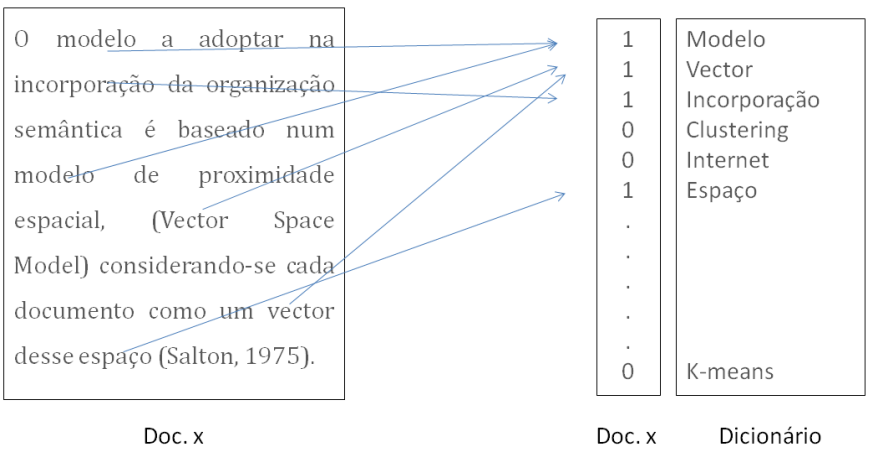


Figura 1: Ilustração do sistema binário para representar um documento através de um vetor.

Os métodos que permitem dar pesos às palavras podem variar. O mais simples é o sistema binário e consiste em verificar se uma palavra de um dicionário aparece no documento, sendo essa palavra representada por 1, ou não aparece no documento, representando-a por 0, tal como se ilustra através do exemplo apresentado na Figura 1.

Por outro lado, o método vetorial Tf tem em conta a frequência com que um determinado termo ocorre no documento. Neste caso, tal como se pode verificar na Figura 2, a coordenada do vetor do documento correspondente à palavra Modelo é 2 porque esta palavra aparece duas vezes no texto. Portanto, neste caso cada coordenada é representada pela frequência com que ocorre no documento.

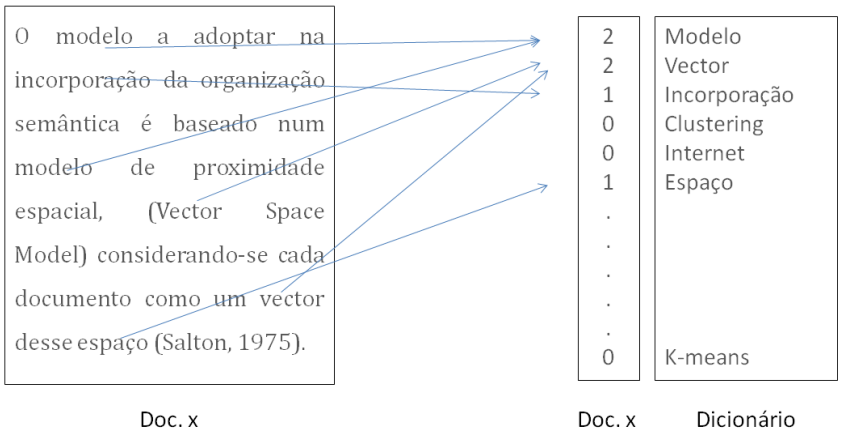


Figura 2: Ilustração do método vetorial Tf.

Idealmente podemos pensar que quanto mais vezes um termo aparece num documento, melhor será para serem formados os *clusters*. O problema é se esse termo aparece em todos os documentos da coleção com frequências similares. Aí deixa de ser um termo relevante para formar os *clusters*. Neste sentido, a utilização do método $Tf - idf$, permite combinar a ocorrência de um determinado termo num documento com a ocorrência desse termo em toda a coleção.

$$Tf - idf = tf \times idf \qquad \text{Equação 1}$$

$$idf = Log\left(\frac{N}{n_i}\right)$$

N: número de documentos

n_i: número de documentos onde aparece o termo p_i

Embora este tipo de representação seja intuitiva e envolva cálculos muito simples, é um processo com um tempo de cálculo bastante demorado quando se usa uma coleção de milhares ou milhões de documentos cada um com milhares de palavras. Para reduzir o

número de palavras a considerar em cada documento é frequente ignorarem-se algumas palavras (as "*stop words*") consideradas sem importância semântica. Este processo pode ser feito identificando essas palavras por comparação com uma lista (de termos) a que se chama "dicionário". Existem dicionários específicos para vários idiomas.

Para além disso, é frequente, associar-se a cada dicionário um "dicionário de sinónimos" incluindo variação de prefixos e sufixos, usando técnicas *Stemming* (Lovins, 1968), por forma a melhor percorrer toda a terminologia do idioma.

A utilização de outras técnicas de redução da dimensão espacial é também muito utilizada, nomeadamente o *Latent Semantic Indexing* (LSI) (Berry & Castellanos, 2008; Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990; Grossman & Frieder, 2004; Konchady, 2006), uma das mais populares. Esta técnica tem por finalidade reduzir a dimensão do espaço vetorial, explorando relações semânticas encobertas através da identificação de padrões entre a utilização das palavras que estão próximas e agrupando os documentos que têm em utilização palavras similares. Assim, documentos que estejam semanticamente relacionados serão agrupados no mesmo *cluster* mesmo que não estejam a utilizar as mesma palavras. O LSI é implementado através de uma técnica da Álgebra linear chamada *Single Value Decomposition* (SVD) (Konchady, 2006; Paige & Saunders, 1981). Contudo, segundo Manning *et al.* (2009), experiências levadas a cabo usando o LSI implementado através do SVD indicaram que a utilização deste tem um custo bastante elevado (nunca houve uma experiência bem sucedida com 1 milhão de documentos). Sendo este um dos maiores obstáculos à difusão do LSI. Para além disso, as experiências também demonstraram que em alguns modos o LSI não conseguiu igualar a eficácia dos índices mais tradicionais.

1.4. Medidas de Similaridade

No modelo de proximidade espacial VSM as funções de similaridade são usualmente baseadas na similaridade/distância entre os vetores que representam cada documento tendo em conta métricas como distância euclidiana, similaridade dos cossenos, Dice, Jaccard, entre outras (A. Huang, 2008; Manning, et al., 2009).

Consideremos $X = (x_1, x_2, \dots, x_n)$ e $Y = (y_1, y_2, \dots, y_n)$ os vetores de dois documentos de texto. Temos:

- Distância Euclidiana

$$D(X,Y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \qquad \text{Equação 2}$$

- Similaridade dos Cossenos

$$Sim(X,Y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} {x_i}^2} \times \sqrt{\sum_{i=1}^{n} {y_i}^2}} \quad \text{Equação 3}$$

- Dice

$$Sim(X,Y) = \frac{2\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} {x_i}^2} + \sqrt{\sum_{i=1}^{n} {y_i}^2}} \quad \text{Equação 4}$$

- Jaccard

$$Sim(X,Y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} {x_i}^2} + \sqrt{\sum_{i=1}^{n} {y_i}^2 - \sum_{i=1}^{n} x_i y_i}} \quad \text{Equação 5}$$

Destas medidas a mais popular é a distância Euclidiana. Contudo, para o *clustering* de documentos de texto é a similaridade dos cossenos a mais utilizada (Feldman & Sanger, 2007).

1.5. Algoritmos de Clustering

Clustering é um processo não supervisionado através do qual uma coleção de documentos é organizada em grupos de documentos similares por meio de um algoritmo (Figura 3).

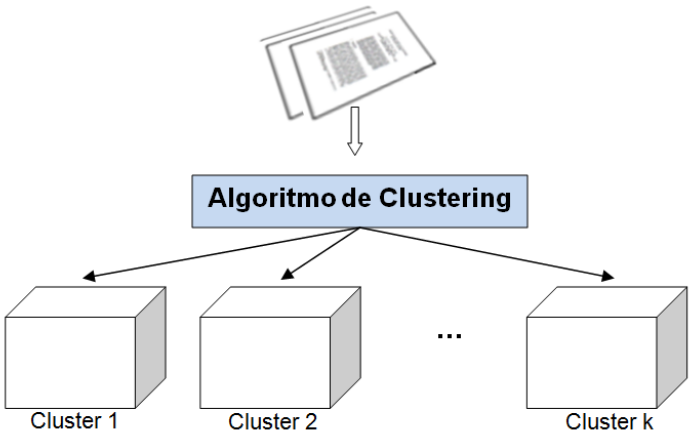


Figura 3: Funcionamento de uma ferramenta de *Clustering*.

A forma como os algoritmos de *clustering* estão organizados em categorias pode variar. Para alguns autores os algoritmos de *clustering* podem fazer parte de uma das seguintes categorias: particionais ou hierárquicos (Feldman & Sanger, 2007), tal como ilustra a Figura 4.

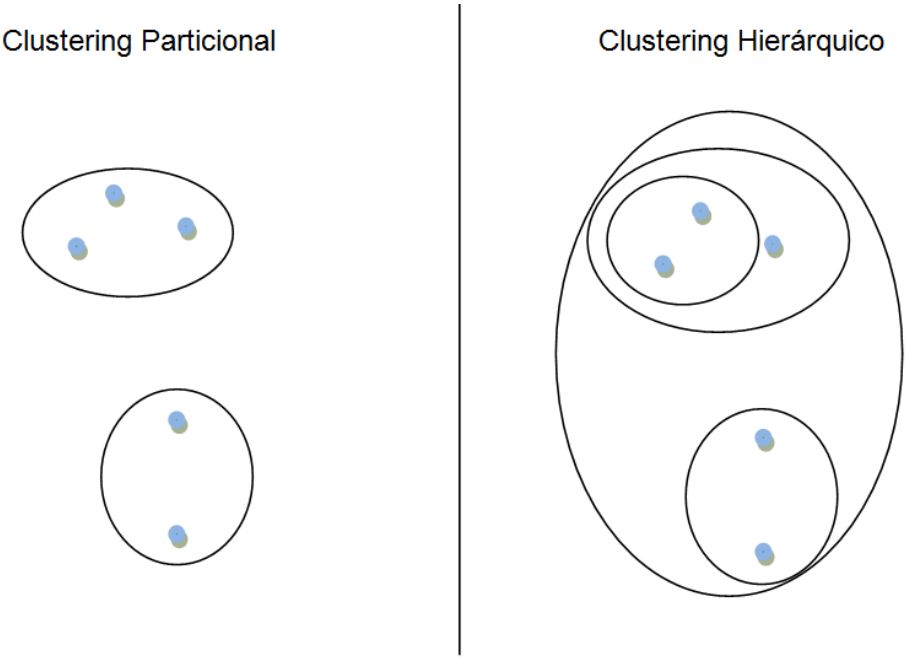


Figura 4: *Clustering* particional e hierárquico

Contudo, por exemplo para Theodorithis e Koutroumbas (2009) os algoritmos de *clustering* podem ser organizados de acordo com as seguintes categorias:

- Sequenciais;
- Otimização da função objetivo;
- Hierárquicos;
- Outros modelos.

Para cada uma das categorias descritas acima serão de seguida apresentados alguns dos algoritmos mais conhecidos.

1.5.1. Algoritmos Sequenciais

Os algoritmos sequenciais produzem um único agrupamento. São considerados métodos bastantes rápidos, já que geralmente os dados são apresentados ao algoritmo uma ou poucas vezes (tipicamente não mais de 6 vezes). O resultado final está dependente da ordem pela qual os vetores são apresentados ao algoritmo. Os *clusters* gerados tendem a ser compactos e em "forma" de elipse ou de circunferência, dependendo da distância utilizada (Theodoridis & Koutroumbas, 2009).

a. *Basic Sequencial Algorithm Scheme* (BSAS)

Segundo Theodoridis e Koutroumbas (2009) a ideia básica do algoritmo é a seguinte: cada novo vetor é associado a um dos *clusters* já existentes ou a um recém-criado, dependendo da distância aos *clusters* formados anteriormente.

Algoritmo

$m = 1$

$C_m = \{x_1\}$

$For\ i = 1\ até\ N$

$Encontra\ C_k: d(x_i, C_k) = min_1 \leq j \leq md(x_i, C_j)$

$Se\ (d(x_i, C_k) > \theta)\ e\ (m > q) então$

- $m = m + 1$
- $C_m = \{x_i\}.$

$Se\ não$

- $C_k = C_k \cup \{x_i\}$
-
- Atualizar o vetor representativo

$End\ \{Se\}$

$End\ \{For\}$

Ilustração do princípio de funcionamento

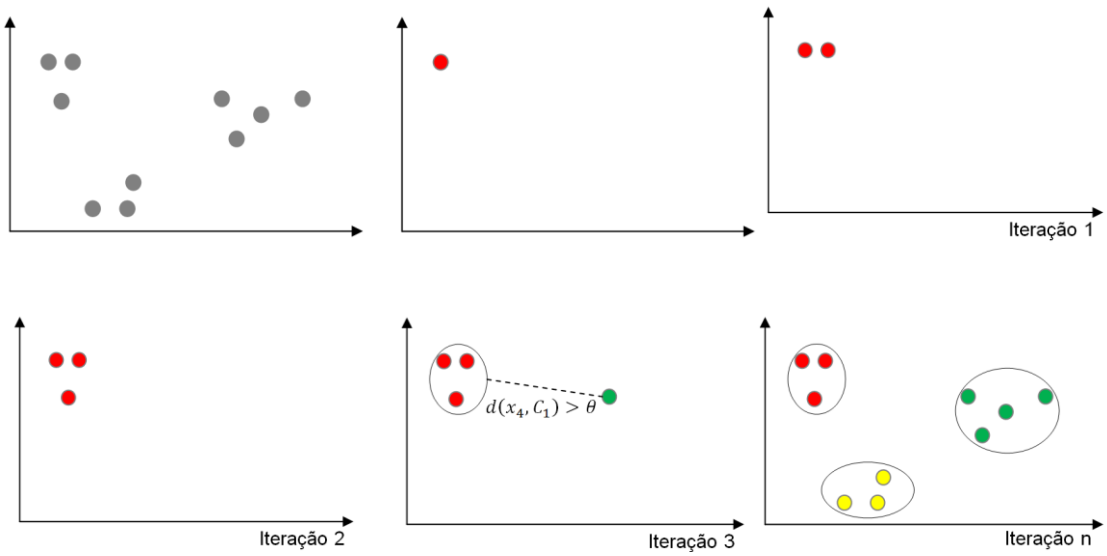


Figura 5: Ilustração do funcionamento do algoritmo BSAS

Complexidade

O algoritmo realiza um único passo para cada dado. Logo a complexidade é $O(n)$.

Este algoritmo está dependente da forma como os dados são apresentados ao algoritmo, alterando a sua ordem pode resultar em diferentes *clusters*.

1.5.2. Algoritmos de *Clustering* Baseados na Otimização de uma Função de Custo

Nesta categoria é definida uma função custo em função da qual o algoritmo é avaliado. Muitos destes algoritmos utilizam conceitos de cálculo diferencial e produzem sucessivos agrupamentos enquanto tentam otimizar a função custo. Estes algoritmos incluem as seguintes subcategorias:

- *Hard* ou *crisp*: cada vetor pertence apenas a um *cluster*.
- *Probabilistic*: refere-se a um tipo especial de algoritmos de *clustering hard* que segue a classificação de argumentos *Bayesian* em que cada vetor é associado ao *cluster* que tem maior probabilidade de pertencer.
- *Fuzzy*: cada vetor pode pertencer simultaneamente a mais do que um *cluster* com diferentes graus de associação.
- *Possibilistic*: mede a possibilidade de um vetor pertencer a um *cluster* específico.
- *Boundary detection*: procura colocar otimamente as fronteiras entre *clusters* ao invés de ter como objetivo distribuir os *cluster* no espaço de uma maneira ótima como é o objetivo dos algoritmos que se situam nas subcategorias descritas anteriormente.

De seguida apresentamos, como exemplo desta categoria, o algoritmo k-means (algoritmo do tipo *hard*) e duas das suas variantes.

a. k-means

O algoritmo *k-means*, apresentado por MacQueen (1967), permite fazer uma partição de um conjunto inicial de documentos (cada documento é representado sob a forma de um vetor) em k *clusters*. O algoritmo é iniciado com a seleção de k sementes. Estas sementes são selecionadas aleatoriamente e de seguida é calculada a distância de cada documento às sementes sendo agrupado à que corresponde menor distância. O processo é repetido até que todos os documentos façam parte de um dos k *clusters*. De seguida, tendo em conta os *clusters* formados é recalculado o centróide (m_k) de cada um dos *clusters*. Cada documento volta a ser associado ao vetor que tem o centróide mais próximo. O processo termina quando não ocorrerem mais alterações.

Portanto, este algoritmo procura minimizar o erro quadrático:

$$\sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - m_k\|^2$$

Algoritmo

1: Seleção de k sementes, ou seja os primeiros centróides.
2: Repetir até que os centróides não sejam alterados
3: Formação dos k *clusters* associando a cada *cluster* os documentos mais próximo ao centróide.
4: Recalcular o centróide de cada *cluster*, média dos pontos presente em cada *cluster*.

A ilustração do seu funcionamento está apresentada na Figura 8.

Análise de Complexidade

A popularidade do algoritmo *k-means* deve-se à sua simplicidade e eficiência.
Segundo Feldman e Sanger (2007) a complexidade de cada interação é $O(kn)$. Apesar do pior caso de complexidade ser mau, em geral é bastante rápido pois o número necessário de iterações é normalmente muito pequeno. Resta ainda referir que, da análise do melhor caso de complexidade, Arthur e Vassilvitskii (2006) mostram que pode exigir no mínimo $2^{\Omega(\sqrt{n})}$ iterações.

Contudo, e segundo os mesmos autores, este algoritmo precisa de um modelo diferente para entender o mundo real. Isto porque a escolha aleatória das sementes iniciais pode resultar num mau exemplo de otimização dos *clusters*.
Uma seleção desadequada de sementes pode dar origem a *clusters* pouco intuitivos. Por exemplo na Figura 6 e na Figura 7, estão representados a tracejado os *clusters* esperados e com linha contínua os que foram efetivamente gerados, pelo que se conclui que estes são de fraca qualidade.

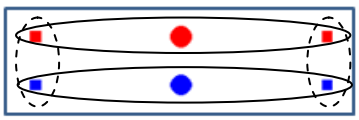


Figura 6: Exemplo 1 de uma seleção desadequada das sementes - adaptado[2] (p. 80).

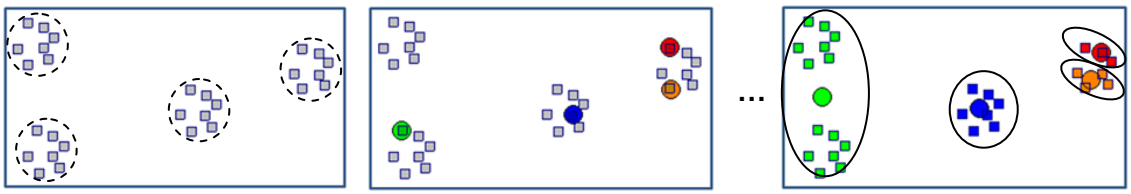


Figura 7: Exemplo 2 de uma seleção desadequada das sementes – adaptado[2] (p. 81).

[2] http://www.learningace.com/doc/2906860/d022dc1331fe7641c4988c34329d3bea/thesis

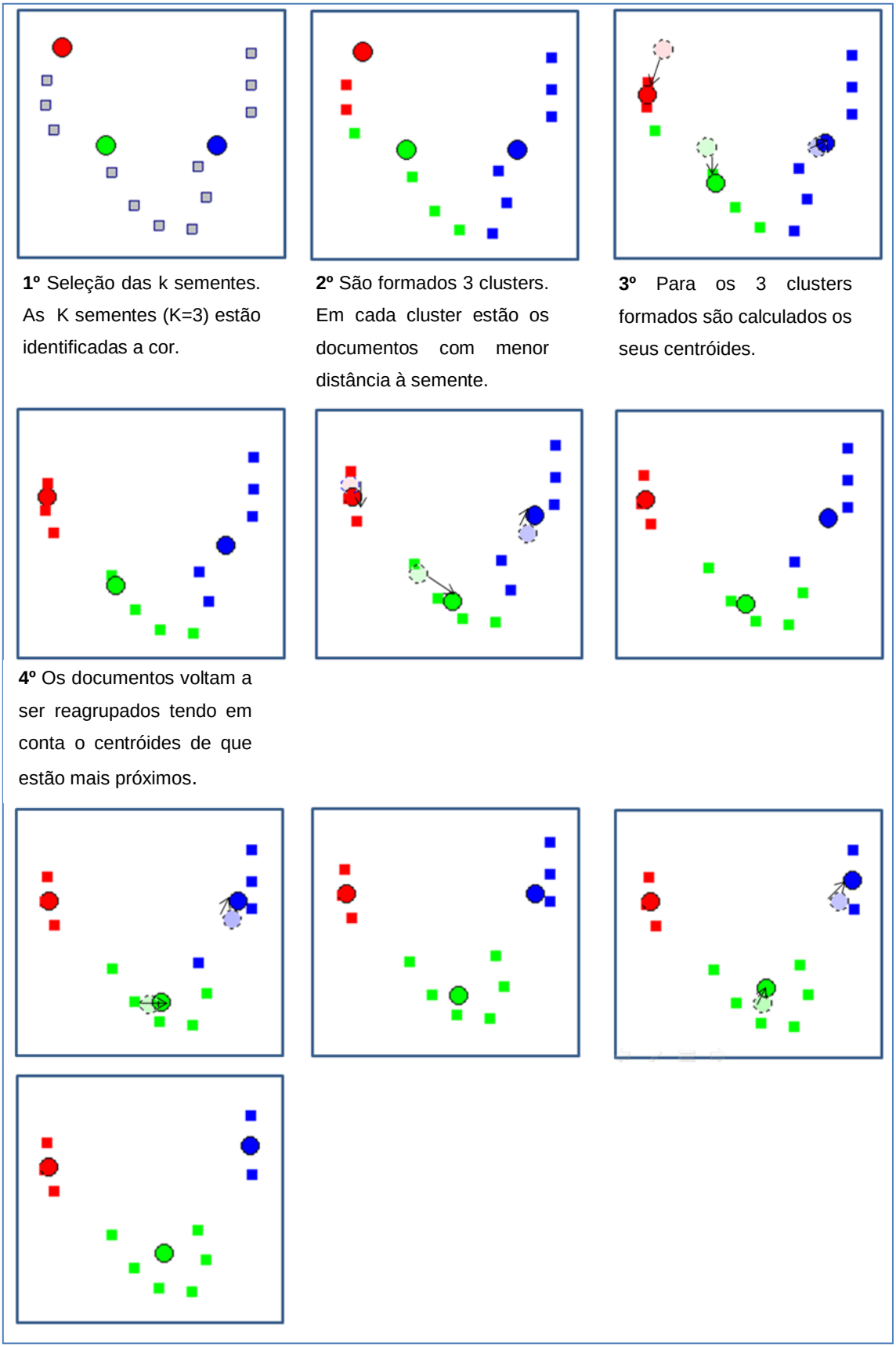


Figura 8: Ilustração do funcionamento do algoritmo *k-means*.

b. k – means ++

O algoritmo *k-means*++ foi proposto por Arthur e Vasilvitskii (2007) no sentido de melhorar a escolha das sementes, permitindo que a sua seleção seja feita com probabilidades específicas.

Algoritmo

1: Seleção do primeiro centróide aleatório e coincidente com um dos pontos $c_1 \in X$

2: para i=2 até k

3: Seleção do próximo centróide $c_i = x' \in X$ com probabilidade $\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$

[$D(x)$ denota a distância entre x e o centróide mais próximo já escolhido.]

4: Executar o algoritmo *k-means*

Outra versão do mesmo algoritmo:

Seleção do conjunto C das k sementes de entre os documentos presentes na coleção

1: Seleção aleatória do primeira semente $c_1 \in X$

2: Para cada x_i, determine-se $D(x_i)$ que é a distância entre x_i e a semente mais próxima já escolhida.

3: Seleção aleatória de um número real y escolhido aleatoriamente entre 0 e $D(x_1)^2 + D(x_2)^2 + \cdots + D(x_n)^2$

4: encontre o inteiro i tal que:

$$D(x_1)^2 + D(x_2)^2 + \cdots + D(x_i)^2 \geq y > D(x_1)^2 + D(x_2)^2 + \cdots + D(x_{(i-1)})^2$$

5: Adicione x_i a C

6: Repetir os passos de 2 a 4 até encontrar as k sementes

Ilustração do algoritmo *k-means*++

1: Seja x_2 a primeira semente aleatória (Figura 9 A).
2: Para cada x_i determine-se a distância a x_2 (Figura 9 B)
3: Seleção aleatória de um número real y entre 0 e $D(x_1)^2 + D(x_2)^2 + \cdots + D(x_n)^2 = 72$. Seja $y = 37$.

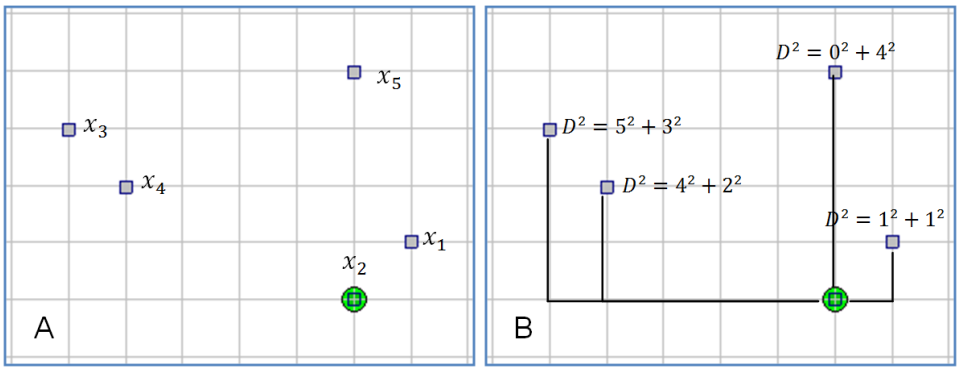


Figura 9: Algoritmo *k-means++*: A) Passo 1 da seleção das sementes; B)Passo 2 da seleção das sementes

4: encontre o inteiro i tal que:

$$D(x_1)^2 + D(x_2)^2 + D(x_3)^2 + D(x_4)^2 \geq y > D(x_1)^2 + D(x_2)^2 + D(x_3)^2$$

$$2 + 0 + 34 + 20 \geq 37 > 2 + 0 + 34$$

Portanto a próxima semente a ser escolhida é x_4 como se pode ver na Figura 10 A.

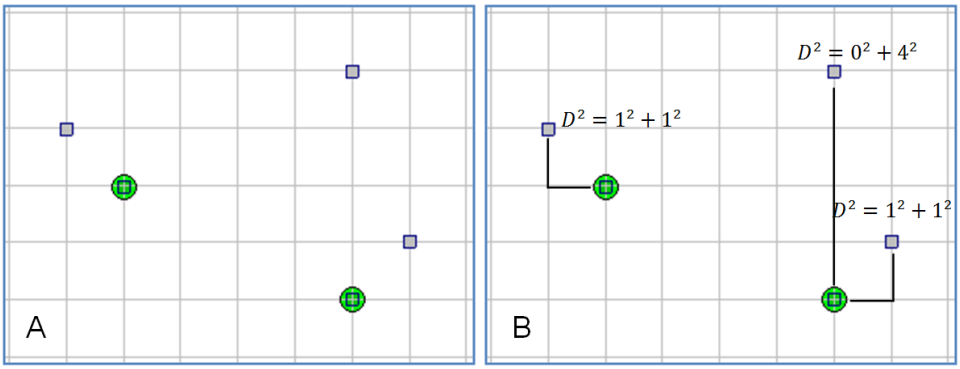


Figura 10: Algoritmo *k-means++*: A) Passo 5 da seleção das sementes; B) Passo 2 da seleção das sementes

Voltamos a selecionar um número aleatório y entre 0 e $D(x_1)^2 + D(x_2)^2 + \cdots + D(x_5)^2 = 20$ (cujos cálculos estão apresentados na Figura 10 B). Seja $y = 9$

Temos que:

$$D(x_1)^2 + D(x_2)^2 + D(x_3)^2 + D(x_4)^2 + D(x_5)^2 \geq y > D(x_1)^2 + D(x_2)^2 + D(x_3)^2 + D(x_4)^2$$

$$2 + 0 + 2 + 0 + 16 \geq 9 > 2 + 0 + 2$$

Portanto a próxima semente a ser escolhida é x_5.

Escolhidas as sementes executa-se o algoritmo *k-means*.

Complexidade *k-means++*

O algoritmo *k-means++* tem complexidade $O(\log k)$ (Arthur & Vassilvitskii, 2007), apresentando uma melhoria quanto ao número de iterações necessárias até atingir a convergência. Arthur e Vassilvitskii (2007) avaliaram experimentalmente o algoritmo *K-means ++* implementando-o e testando-o em C++ e concluíram que a seleção inicial das sementes melhora substancialmente a execução e a precisão do algoritmo *k-means*.

c. Single Pass seed Selection (SPSS)

No algoritmo *k-means*++ a primeira semente é selecionada aleatoriamente no conjunto de pontos disponíveis, o SPSS (Pavan, Rao, Rao, & Sridhar, 2010) seleciona a semente inicial de acordo com o ponto que está mais próximo de todos os outros.

O SPSS assume que os n pontos são distribuídos uniformemente pelos k *clusters* e por isso espera-se que cada *cluster* tenha n/k pontos.

Algoritmo

Seja C o conjunto das k sementes iniciais.

1: Cálculo da matriz das distâncias entre todos os pontos

2: Soma das distâncias de cada ponto a todos os pontos

3: A primeira semente é o ponto que está mais próximo de todos os outros pontos

4: Adicionar a C a primeira semente

5: Para cada x_i, determine-se $D(x_i)$ que é a distância entre x_i e a semente mais próxima já escolhida.

6: Seja y a soma das distâncias dos primeiros n/k pontos mais próximos da semente já escolhida

7: encontre o inteiro i tal que:

$$D(x_1)^2 + D(x_2)^2 + \cdots + D(x_i)^2 \geq y > D(x_1)^2 + D(x_2)^2 + \cdots + D(x_{(i-1)})^2$$

5: Adicione x_i a C

6: Repetir os passos de 5 a 7 até encontrar as k sementes.

Considere-se o exemplo apresentado na Figura 11 onde se ilustrará a execução do algoritmo.

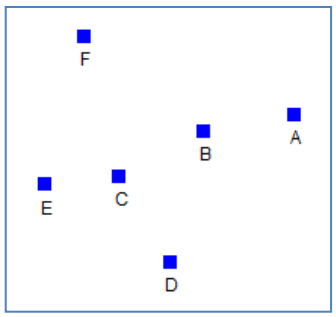


Figura 11: Representação de 6 documentos usando um dicionário com duas palavras.

Em primeiro lugar será feito o cálculo da matriz das distâncias entre os pontos. Pelo algoritmo SPSS a primeira semente é C, porque é a que está mais próxima de todos os outros pontos, como se pode confirmar pela matriz das distâncias (Tabela 2).

Tabela 2: Cálculo da matriz das distâncias entre todos os pontos.

	A	B	C	D	E	F	Soma das distâncias
A	0	1,59	3,26	3,36	4,51	3,95	16,67
B	1,59	0	1,7	2,37	2,94	2,72	11,32
C	3,26	1,7	0	1,77	1,28	2,54	10,55
D	3,36	2,37	1,77	0	2,58	4,26	14,34
E	4,51	2,94	1,28	2,58	0	2,7	14,01
F	3,95	2,72	2,54	4,26	2,7	0	16,17

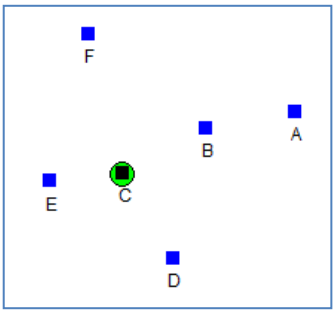


Figura 12: Seleção da primeira semente – algoritmo SPSS

De seguida, é necessário encontrar y que é a soma das distâncias dos primeiros 2 pontos (6/3) mais próximos de C, isto significa que $y = 1{,}28 + 1{,}7 = 2{,}98$

$$1{,}28^2 < 2{,}98 \leq 1{,}28^2 + 1{,}7^2$$

$$1{,}63 < 2{,}98 < 1{,}63 + 2{,}89$$

Logo a próxima semente é B, tal como se ilustra na Figura 13.

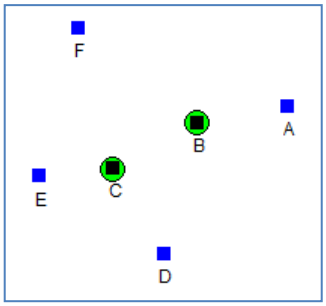


Figura 13: Seleção da segunda semente – algoritmo SPSS.

Observando a Tabela 2 constatámos que os pontos mais próximos da semente B distam desta 1,59 e 1,7 e por isso temos:

$$y = 1{,}59 + 1{,}7 = 3{,}29$$

$$1{,}7^2 < 3{,}29 \leq 1{,}7^2 + 2{,}37^2$$

$$2{,}89 < 3{,}29 \leq 2{,}89 + 5{,}62$$

Portanto, a próxima semente é D.

Síntese – Algoritmo *k-means*

A escolha inicial das sementes é efetivamente um dos grandes problemas do *algoritmo k-means*. Para além dos algoritmos apresentados existem outros que têm vindo a ser apresentados no sentido de otimizar os *clusters* criados, como também é o caso do algoritmo ISO-DATA que funde os *clusters* no caso da distância entre os seus centroides ser inferior a um certo limite, separando os *clusters* com variância excessiva (Feldman & Sanger, 2007).

Outro dos grandes problemas do algoritmo *k-means* está na escolha do k. Uma das possibilidades para determinar k consiste em executar o algoritmo várias vezes para diferentes valores de k, sendo escolhido o que apresentar melhores resultados para alguma função de qualidade (como por exemplo o coeficiente de *Silhouette* que descrevemos de seguida). Contudo, este método implica um elevado custo do ponto de vista computacional, mercê do elevado número de cálculos.

Para além dos problemas já apresentados também é reconhecido que é muito sensível aos *outliers* e ao ruído (Theodoridis & Koutroumbas, 2009).

d. Coeficiente de *Silhouette*

A decisão sobre o número de *clusters* pode ter em consideração a análise do coeficiente de *Sillhouette* (Theodoridis & Koutroumbas, 2009). Esta medida permite analisar a qualidade dos *clusters* de $k = 2 \dots n$, sendo n o número de documentos, e desta forma determinar o número de *clusters* adequado aos documentos que devem ser agrupados.
O cálculo do i-ésimo elemento é determinado pela equação:

$$S(i) = \frac{b_i - a_i}{\max(a_i, b_i)} \qquad \text{Equação 6}$$

a_i: distância média entre o *i-ésimo* elemento do *cluster* e os restantes elementos

b_i: distância mínima entre o i-ésimo elemento do grupo e os restantes elementos que não pertencem ao *cluster*

1.5.3. Algoritmos Hierárquicos

Os algoritmos hierárquicos podem ser aglomerativos ou divisivos. Os algoritmos aglomerativos começam por colocar cada objeto num *cluster* separado e sucessivamente são fundidos os *clusters* até se obter um único *cluster* geral. Os algoritmos divisivos iniciam com um único *cluster* que contém todos os objetos e sucessivamente esse *cluster* é separado noutros *clusters* até que cada objeto esteja associado a um *cluster* individual (Feldman & Sanger, 2007).

No exemplo apresentado, podemos ver como se podem fundir os vários *clusters* até se obter um único *cluster*. Assim, inicialmente temos cada objeto associado a um único *cluster*: $\{1\},\{2\},\{3\},\{4\},\{5\}$. De seguida, $\{1\}$ e $\{3\}$, por apresentarem a maior similaridade, são fundidos num único *cluster* e obtém-se $\{1,3\},\{2\},\{4\},\{5\},\{6\}$. Os próximos *cluster*s a serem fundidos são o $\{2\}$ e o $\{5\}$ como se pode verificar através do dendrograma que exibe as várias fusões entre os *cluster*s (Figura 14).

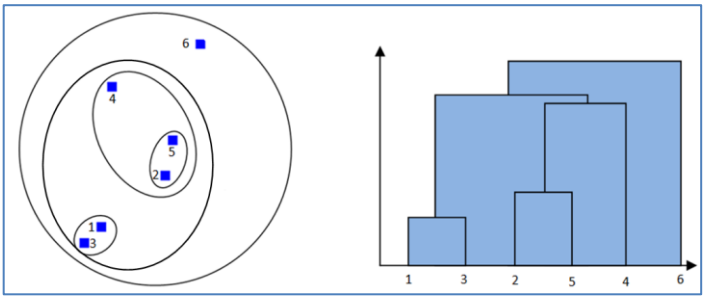


Figura 14: Ilustração da construção de um dendrograma

***Hierarchical Agglomerative Clustering* (HAC)**

Cada documento é colocado num *cluster* separado e, sucessivamente, são formados novos pares de *cluster*s de entre os que são mais similares. O algoritmo termina quando todos os documentos são incluídos num único *cluster*.

Algoritmo

1: Calcula a matriz das distâncias entre os vários objetos
2: Cada objeto é um *cluster*
3: Repete
4: Os dois *clusters* mais próximos são fundidos
5: atualiza a matriz das distâncias
6: até restar um único *cluster*

Diferentes versões do algoritmo podem ser apresentadas, pois está dependente do critério de similaridade escolhido. Assim, para serem determinados os *clusters* mais próximos podem ser usados os seguintes métodos: *single-link*; *complete-link*; *average-link*; *center of gravity* e *Ward's Method* (Feldman & Sanger, 2007).

a. Single-link

No caso do *single link,* a distância entre dois *clusters* é determinada pela distância mínima entre os seus elementos (Equação 7) (Manning, et al., 2009).
Como se pode observar na Figura 15 a distância entre os *clusters* é definida pela

distância entre os elementos mais próximos de cada *cluster*. Assim podemos ver que o *cluster* que se encontra na parte superior esquerda está mais próximo do *cluster* que está na parte superior direita (comprimento da linha contínua) do que do *cluster* que está na parte inferior esquerda (comprimento da linha a tracejado).

$$D(C_i, C_j) = \min \{d(x, y): x \in C_i, y \in C_j\} \qquad \text{Equação 7}$$

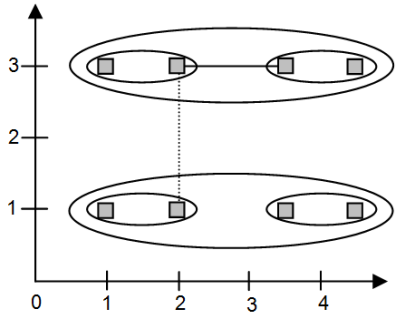


Figura 15: Ilustração do método Single link adaptado de Manning et al. (2009, p. 382).

Considere-se a Figura 16 para apresentar mais um exemplo de *clustering* hierárquico usando o método *Single Link*.

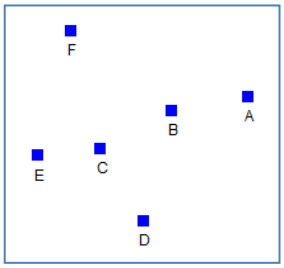


Figura 16: Representação de 6 documentos usando um dicionário com duas palavras.

Em primeiro lugar é necessário calcular a matriz das distâncias (usando a distância Euclidiana) entre os *clusters* individuais (Tabela 3).

Tabela 3: Matriz das distâncias entre os *clusters* – primeira iteração.

Clusters	{A}	{B}	{C}	{D}	{E}	{F}
{A}	0	1,59	3,26	3,36	4,51	3,95
{B}		0	1,7	2,37	2,94	2,72
{C}			0	1,77	1,28	2,54
{D}				0	2,58	4,26
{E}					0	2,7
{F}						0

Na Tabela 3 observamos que os *clusters* que estão mais próximos são {C} e {E} e por isso são fundidos num único *cluster*.

De seguida recalculámos a matriz das distâncias tal como se apresenta na Tabela 4.

Tabela 4: Matriz das distâncias entre os *clusters* – segunda iteração.

Clusters	{A}	{B}	{C,E}	{D}	{F}
{A}	0	1,59	3,26	3,36	3,95
{B}		0	1,7	2,37	2,72
{C,E}			0	1,77	2,54
{D}				0	4,26
{F}					0

Isto significa que os próximos *clusters* a serem fundidos são $\{A\}$ e $\{B\}$ e volta a ser recalculada a matriz das distâncias (Tabela 5).

Tabela 5: Matriz das distâncias entre os *clusters* – terceira iteração.

Clusters	{A,B}	{C,E}	{D}	{F}
{A,B}	0	1,7	2,37	2,72
{C,E}		0	1,77	2,54
{D}			0	4,26
{F}				0

Isto significa que $\{A, B\}$ e $\{C, E\}$ passam a fazer parte do mesmo *cluster*. O processo repete-se até que todos os pontos façam parte do mesmo *cluster*. Na Figura 17 pode observar-se a construção dos vários *clusters* através do dendrograma.

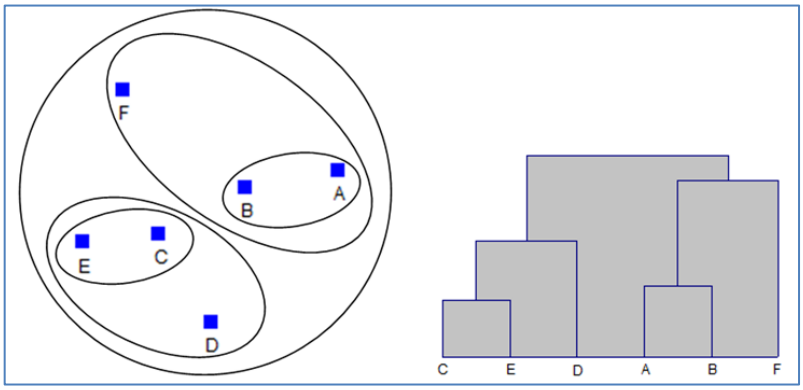


Figura 17: Ilustração da construção de um dendrograma usando o método *Single Link*.

Este método é bastante sensível a ruído e *outliers*, pois basta que exista um par de elementos muito próximo, apesar dos restantes poderem estar muito afastados.

b. Complete-link

No método *Complete-link* a distância entre dois *clusters* é determinada pela distância máxima entre os seus elementos (Equação 8) (Manning, et al., 2009).

$$D(C_i, C_j) = \max\{d(x, y) : x \in C_i, y \in C_j\} \qquad \text{Equação 8}$$

Considere-se o exemplo apresentado abaixo

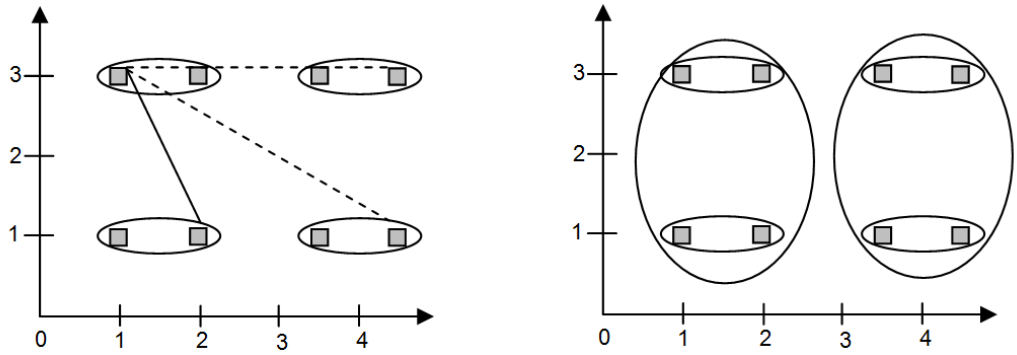


Figura 18: ilustração do método *Complete-Link* adaptado de Manning et al. (2009, p. 382).

Na figura da esquerda são comparadas as distâncias entre o *cluster* superior esquerdo e os restantes *clusters*. Podemos assim constatar que os *cluster* mais próximo é o *cluster* inferior esquerdo.

Considere-se o mesmo exemplo apresentado na Figura 16 para ilustrar o método *Complete-link*. Assim, na Figura 19, pode observar-se que as duas primeiras fusões são iguais nos dois métodos. De seguida, verifica-se que {D} e {E,C} são os *clusters* que estão mais próximos. Finalmente {F} funde-se com {A,B} porque a distância de {F} a {A} é menor do que a distância de {F} a {D}. Note-se que neste método considera-se que a distância entre dois *clusters* é a distância máxima entre os seus elementos.

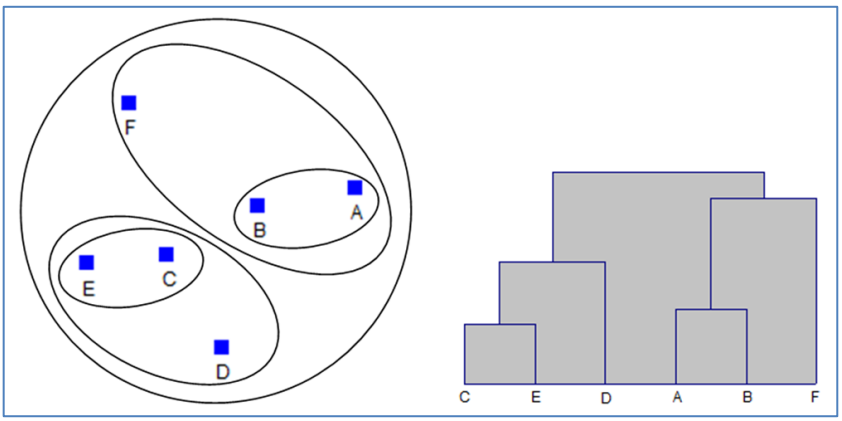


Figura 19: Ilustração da construção de um dendrograma usando o método *Complete-Link*.

c. *Center of gravity*

No método *center of gravity* a distância entre dois *clusters* é determinada pela distância entre os centróides (Feldman & Sanger, 2007). Por exemplo, na Figura 20, podemos ver que o primeiro *cluster* é formado pelo *cluster* {E} e pelo *cluster* {C}, sendo que a partir deste momento a distância dos restantes *clusters* ao *cluster* {E,C} será calculada ao seu centróide.

$$D_{centróides}(C_i, C_j) = d(r_i, r_j) \qquad \text{Equação 9}$$

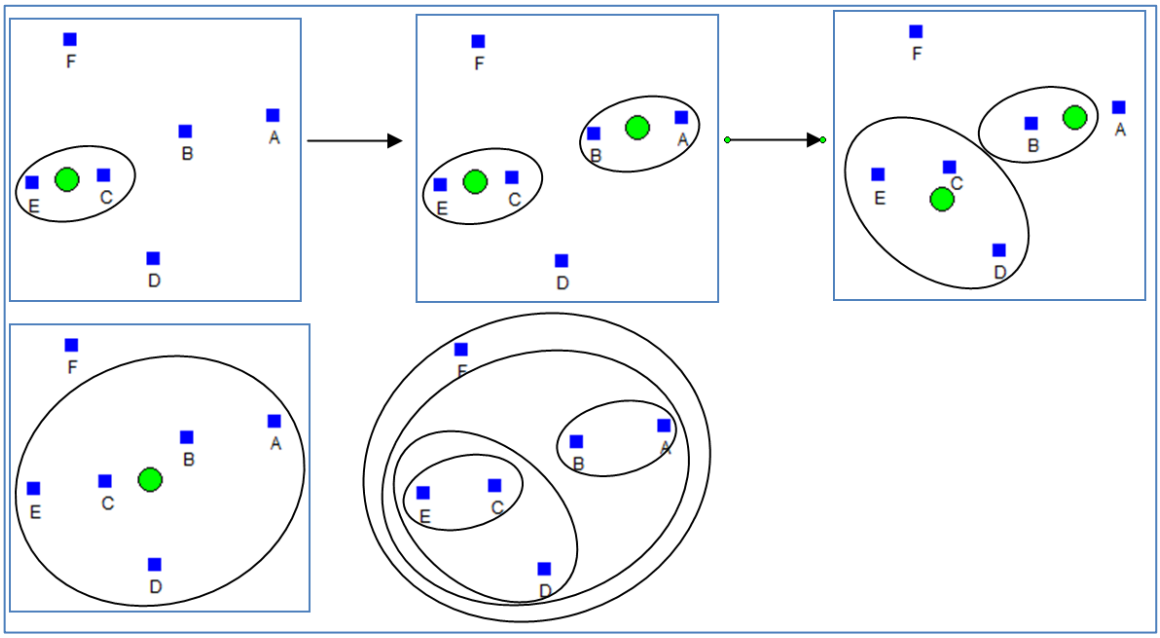


Figura 20: Ilustração do método *center of gravity*.

d. *Average link*

No método *average link* a distância entre dois *clusters* é determinada pela média das distâncias entre todos os pares de pontos entre os dois *clusters* (Feldman & Sanger, 2007).

$$D_{average\_link}(C_i, C_j) = \frac{1}{|C_i| \times |C_j|} \sum_{x \in C_i, y \in C_j} d(x, y) \qquad \textbf{Equação 10}$$

Em primeiro lugar determina-se a matriz da distância entre todos os *clusters* individuais (Tabela 6).

Analisando a matriz das distâncias (Tabela 6), concluímos que {C} e {E} passam a fazer parte do mesmo *cluster*: $\{C, E\}, \{A\}, \{B\}, \{D\}, \{E\}$.

Como a distância entre os *clusters* é determinada pela média das distâncias entre todos os pares de pontos que constituem os dois *clusters* vamos determinar a média das distâncias entre o *cluster* $\{C, E\}$ e os *clusters* $\{A\}, \{B\}, \{D\}$ e $\{F\}$ (Tabela 7).

Tabela 6: Matriz das distâncias entre todos os *clusters* – primeira iteração – *Average link*.

Clusters	{A}	{B}	{C}	{D}	{E}	{F}
{A}	0	1,59	3,26	3,36	4,51	3,95
{B}		0	1,7	2,37	2,94	2,72
{C}			0	1,77	1,28	2,54
{D}				0	2,58	4,26
{E}					0	2,7
{F}						0

Tabela 7: Matriz das distâncias entre todos os *clusters* – segunda iteração – *Average link*.

Clusters	{C,E}	{A}	{B}	{D}	{F}
{C,E}	0	3,9	2,32	2,2	2,62
{A}		0	1,59	3,36	3,95
{B}			0	2,47	2,72
{D}				0	4,26
{F}					0

$$d_{\{C,E\},\{A\}} = \frac{3,26 + 4,51}{2} = 3,9$$

$$d_{\{C,E\},\{B\}} = \frac{1,7 + 2,94}{2} = 2,32$$

$$d_{\{C,E\},\{D\}} = \frac{1,77 + 2,58}{2} = 2,2$$

$$d_{\{C,E\},\{F\}} = \frac{2,54 + 2,7}{2} = 2,62$$

Portanto os *clusters* {A} e {B} vão fundir-se já que apresentam a menor média e obtemos os *clusters*: {C,E}, {A,B}, {D} e {F}. Prosseguimos com o cálculo das distâncias entre o *cluster* {A,B} e os restantes *clusters* (Tabela 8).

Tabela 8: Matriz das distâncias entre todos os *clusters* – terceira iteração – *Average link*.

Clusters	{C,E}	{A,B}	{D}	{F}
{C,E}	0	3,1	2,2	2,62
{AB}		0	2,9	3,3
{D}			0	4,26
{F}				0

$$d_{\{C,E\},\{A,B\}} = \frac{3,26 + 1,7 + 4,51 + 2,94}{4} = 3,1$$

$$d_{\{A,B\},\{D\}} = \frac{3,36 + 2,37}{2} = 2,9$$

$$d_{\{A,B\},\{F\}} = \frac{3,95 + 2,72}{2} = 3,3$$

Assim, a nova fusão passa a ser {C,E,D} e é necessário calcular a distância média entre este novo *cluster* e cada um dos restantes cujos resultados se apresentam na Tabela 9.
O próximo passo consiste em fundir os *clusters* {C,E,D} e {A,B} e finalmente o resultante desta fusão com o *cluster* {F}. Na Figura 21 ilustramos todas as fusões descritas.

Tabela 9: Matriz das distâncias entre todos os *clusters* – quarta iteração – *Average link.*

Clusters	{C,E,D}	{A,B}	{F}
{C,E,D}	0	3,1	2,62
{AB}		0	3,3
{F}			0

$$d_{\{C,E,D\},\{A,B\}} = \frac{3{,}26 + 1{,}7 + 4{,}51 + 2{,}94 + 3{,}36 + 2{,}27}{6} = 3{,}02$$

$$d_{\{C,E,D\},\{F\}} = \frac{2{,}54 + 2{,}7 + 4{,}26}{3} = 3{,}2$$

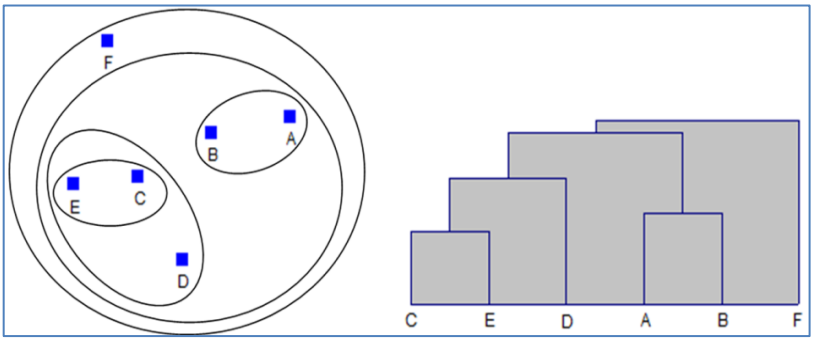


Figura 21: Ilustração do método *Average Link.*

e. Ward's Method

O *Ward's Method* baseia-se na análise da variância (Theodoridis & Koutroumbas, 2009). Assim, parte da soma dos quadrados dos erros de cada novo *cluster* em relação ao centróide. O método tem por finalidade analisar todos os possíveis pares de *clusters* visando detetar qual das fusões produz um aumento menor da soma dos quadrados dos erros.

$$D_w(C_i, C_j) = \sum_{x \in C_i}(x - r_i)^2 + \sum_{x \in C_j}(x - r_j)^2 - \sum_{x \in C_{ij}}(x - r_{ij})^2 \qquad \text{Equação 11}$$

r_i: *centróide do cluster* C_i
r_j: *centróide do cluster* C_j
r_{ij}: *centróide do cluster* C_{ij}

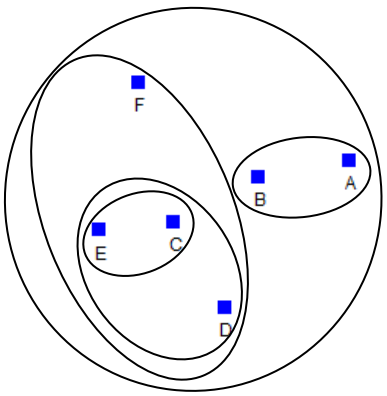


Figura 22: Ilustração do *Ward's Method*.

Complexidade do HAC

A complexidade do HAC é $O(n^2 s)$, onde n é o número de pontos e s a complexidade resultante do cálculo da similaridade entre os *clusters* (Feldman & Sanger, 2007).

Na primeira iteração, todos os métodos HAC necessitam de calcular a similaridade de todos os pares de n instâncias individuais o que corresponde a $O(n^2)$.

Em cada uma das subsequentes n-2 iterações de fusão, tem de calcular a distância entre o *cluster* criado mais recentemente e todos os restantes *clusters*.

Portanto, para manter a matriz de similaridade em $O(n^2)$, o cálculo de similaridade com qualquer *cluster* deve ser feito num tempo constante.

1.5.4. Outras Abordagens

Nesta categoria estão incluídos os algoritmos que utilizam técnicas de *clustering* que não se enquadram nas categorias anteriores: técnicas baseados em grafos, redes neurais artificiais, em densidade (algoritmo DBSCAN, exemplo que apresentaremos de seguida) entre outros.

a. DBScan – *Density Based Spacial Clustering of Applications with Noise*

O algoritmo DBScan (Sander, Ester, Kriegel, & Xu, 1998) é um algoritmo baseado na densidade dos pontos a agrupar. Os *clusters* são formados por regiões densas de pontos e separados entre si por regiões de baixa densidade.

Seja o conjunto:

$$N_\varepsilon(P) = \{Q \in D | d(P,Q) \leq \varepsilon\}, P \neq Q$$

Chamado de ε vizinhança de P e que é constituída por todos os pontos do repositório D que estão a uma distância ε de P. O cardinal deste conjunto é notado por $|N_\varepsilon(P)|$.

A formação dos *clusters* envolve os conceitos de *alcançável por densidade* e *conetividade por densidade* que serão definidos de seguida.

Assim, diz-se que um ponto P é *diretamente alcançável por densidade* a partir de outro ponto Q se:

1. $P \in N_\varepsilon(Q)\ e$
2. $|N_\varepsilon(P)| \geq MinPts$

Isto significa que o ponto P tem de pertencer à vizinhança de Q e simultaneamente tem de ser um ponto interior (condição dada no ponto 2), ou seja, um ponto é interior se o cardinal da vizinhança de P for superior a um determinado número de pontos ($MinPts$).

Daqui decorre que dois pontos só são mutuamente *diretamente alcançáveis por densidade* se ambos forem pontos interiores. Na Figura 23 podemos constatar que P é *diretamente alcançável por densidade* a partir de Q porque P pertence à vizinhança de Q e porque P é um ponto interior, ainda que Q não seja diretamente alcançável por densidade a partir P porque Q não é um ponto interior.

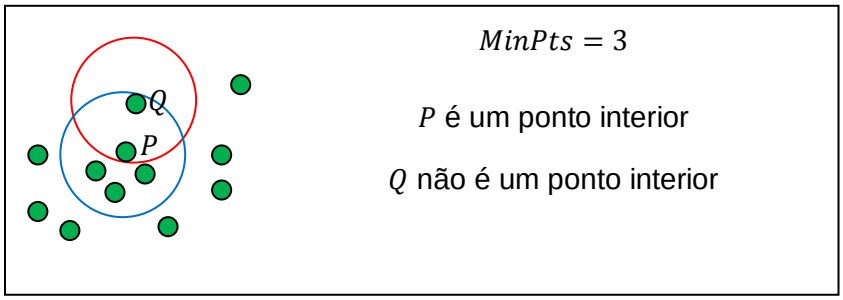


Figura 23: Dois pontos que não são mutuamente diretamente alcançáveis por densidade.

Os pontos que não são interiores são pontos de borda ou *outliers*. Os pontos de borda são os que podem ser alcançáveis por densidade a partir de outros pontos interiores e os *outliers* são os restantes. Surge então a necessidade de definir quando um ponto pode ser *alcançável por densidade* a partir de outro ponto.

Um ponto p é *alcançável por densidade* a partir de um ponto q se existe uma cadeia de pontos $P, \dots, P_n$, $P_1 = Q$, $P_n = P$ tal que P_{i+1} é *diretamente alcançável por densidade* a partir de P_i. Como se observa na Figura 24, existe uma cadeia de pontos, Q, S, P tal que P é diretamente alcançável por densidade a partir de S e S é diretamente alcançável por densidade a partir de Q logo por transitividade P é alcançável por densidade a partir de Q.

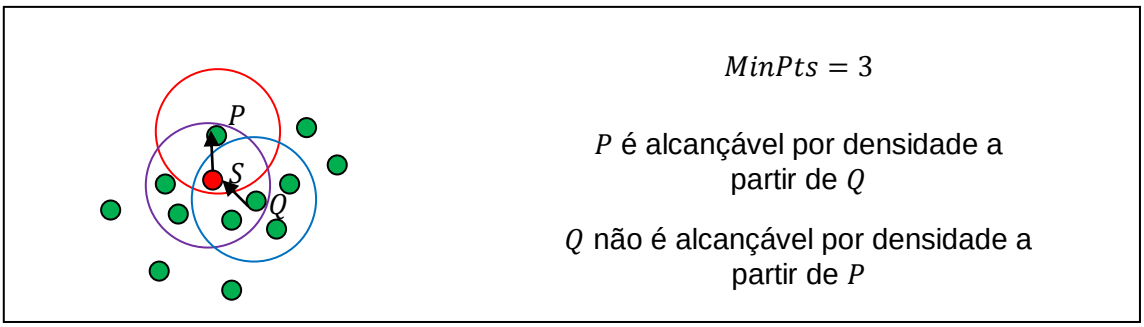


Figura 24: pontos *alcançáveis por densidade*.

Note-se ainda que dois pontos de borda pertencentes a um mesmo *cluster* podem não ser alcançáveis por densidade um a partir do outro. Neste sentido, é introduzido um novo conceito: *conectados por densidade*.

Assim, um ponto P diz-se *conectado por densidade* a um ponto Q se existir um ponto R tal que ambos – P e Q – sejam alcançáveis por densidade a partir de R, tal como ilustra a Figura 25.

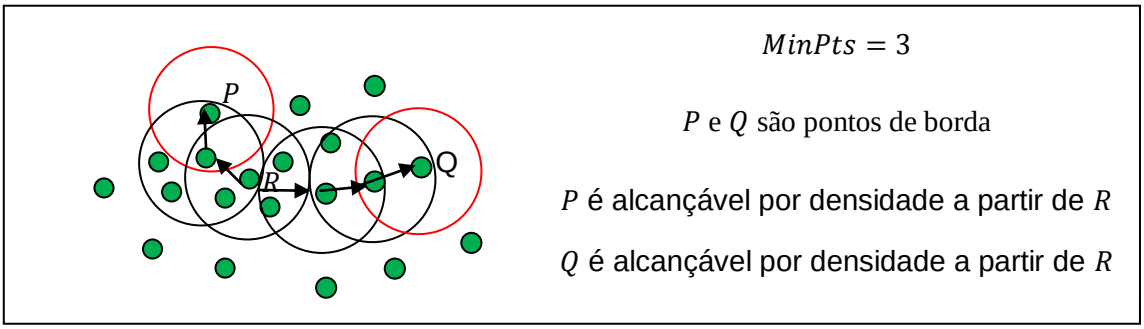


Figura 25: pontos *conectados por densidade*.

Neste algoritmo dois pontos fazem parte do mesmo *cluster* se estiverem conectados por densidade.

Algoritmo

1: Escolhe-se um ponto P arbitrariamente.

2: Selecionam-se todos os pontos que são *alcançáveis por densidade* a partir de P $(\varepsilon, MinPts)$.

3: Se P for um ponto interior, forma-se um grupo.

4: Se P for um ponto fronteira, visitar próximo ponto.

5: Repetir o processo até que todos os pontos sejam analisados.

Complexidade

Segundo os autores uma vez que se espera que as ε-vizinhanças sejam pequenas em comparação com todo o repositório, a complexidade média da consulta de uma única região tem tempo de execução O (log n). Para cada um dos n pontos do repositório, há no máximo uma consulta por região. Assim, a média complexidade DBSCAN em termos de tempo de execução é O (n * log n) (Ester, Kriegel, Sander, & Xu, 1996).

No entanto, sem o uso de uma estrutura de indexação, a complexidade de tempo de execução é *O* (n^2) (Sander, et al., 1998).

Este algoritmo apresenta algumas vantagens, por exemplo face ao algoritmo *k-means*, consegue formar *clusters* com formas arbitrárias; é robusto para o ruído e não precisa de um k à priori. Contudo, requer regiões conectadas com suficiente densidade (falha se a matriz for muito esparsa), funciona mal em repositórios com variação de densidade e é menos eficiente que o algoritmo *k-means*. Para além disso é sensível à escolha dos parâmetros ε e MinPts.

1.6. Síntese

Com a revisão de literatura feita neste capítulo pretendemos selecionar um algoritmo de *clustering* no sentido de integrarmos o *tagging* social no processo de agrupamento automático. A nossa escolha recaiu sobre o algoritmo *k-means* porque é um algoritmo consensualmente considerado eficiente e que também tem sido utilizado em estudos similares. Por exemplo, foi utilizado num estudo feito por Ramage et al. (2009), com o objetivo de integrar as *tags* geradas pelos utilizadores de grupos sociais de larga escala, como no caso do *del.icio.us*, visando averiguar se o agrupamento automático de páginas *Web* em *clusters* com proximidade semântica é melhorado.

Para além disso, além de idealizarmos a integração das *tags* no VSM (tal como fez Ramage, mas em função da ocorrência das *tags* no documento de texto), o algoritmo *k-means* adequa-se à implementação de uma nova forma de selecionar as sementes iniciais, utilizando a rede de *tags* como elemento do processo.

Capítulo 2
Da Web 2.0 ao Tagging Social

Sabendo que frequentemente nos referimos à sociedade atual como “consumista”, não devemos esquecer a sua vertente produtora. O século XXI viu amadurecer a vontade de produzir, de criar, de partilhar com os demais, ideias, textos, imagens, enfim, tudo o que outrora estaria restrito a um círculo reduzido (excetuando aqueles que ocupassem cargos destinados aos fazedores de opinião) na medida em que proporcionou a todos a hipótese de o fazer facilmente.

A evolução da *World Wide Web* levou ao surgimento de novos conceitos como o da *Web 2.0* e *Web* Social. Segundo Tim O’ Reilly (2007):

> *Web* 2.0 é a rede como plataforma, abarcando todos os dispositivos conectados; As aplicações da *Web* 2.0 são aquelas que realizam a maioria das vantagens intrínsecas a essa plataforma: distribuiem o *software* como um serviço continuamente atualizado que melhora quanto mais pessoas o usarem, utilizando e misturando dados de múltiplas fontes, incluindo utilizadores individuais, enquanto oferece simultaneamente os seus próprios dados e serviços de forma a permitir a “remixagem” por outros, criando efeitos de rede através de uma "arquitetura de participação" e indo além da metáfora da página da *Web* 1.0 para proporcionar experiências ricas aos utilizadores (p. 17).

Portanto, a web 2.0 caracteriza-se por disponibilizar um conjunto de aplicações que permitem ao utilizador a fácil publicação, edição e partilha de conteúdos. Disso são exemplo os chats, os fóruns, depois os blogues, as redes sociais, os sites de partilha de música, vídeo e fotos. Acima de tudo, partilha de conhecimento, presente em muitas das suas vertentes (desde a informação pessoal à global, desde o blogue que cobre as

notícias do bairro à *Wikipédia*, que aspira a concentrar conhecimento). O conceito da *Web* Social surge precisamente da importância que é dada à interação social entre os utilizadores da web 2.0 que segundo Kaplan e Haenlein (2010) se define como "um grupo de aplicações baseadas na internet sutendadas pelos fundamentos tecnológicos e ideológicos da *web* 2.0, que permitem a criação e a troca de conteúdos gerados pelos utilizadores" (p. 61).

Assim, a interoperabilidade e a cooperação são, de acordo com Maslov *et al.* (2009) conceitos intimamente associados a "uma rica e interligada mescla de aplicações Web e fornecedores de informação unidos por um conjunto de protocolos, *open-standards* e acordos de cooperação" (p. 3), que permitem esta cultura de abertura e cooperação, fomentando a inteligência coletiva e a partilha do conhecimento.

Contudo, da participação massiva dos utilizadores advém um crescente fluxo de informação que tem obrigado à criação de novas técnicas de gestão de pesquisa e de acesso à informação (Lee, Goh, Razikin, & Chua, 2009).

As dinâmicas criadas entre os utilizadores da *Web* Social vêm, de uma forma natural, proporcionar formas interessantes de auxiliar na organização da informação ao serem criadas "folksonomias". O termo *folksonomy* (Wal, 2007) foi criado por Vander Wal e deriva da aglutinação dos termos *folk* (povo) e *taxonomy* (taxonomia). "Folksonomias" surgem naturalmente quando um conjunto de utilizadores interessados numa determinada informação decide descrevê-la através de comentários ou da atribuição de *tags*, fornecendo elementos importantes para categorizar essa informação.

Exemplos de iniciativas como a levada a cabo pela *Library of the Congress*[3] ou o *"Project Steve"*[4], mostram o poder que reside na criação de uma "folksonomia*"*.

As potencialidades do *tagging* tornaram-se evidentes quando a *Library of Congress* lançou um projeto piloto no *Flickr*, um popular Web site de partilha de fotografias.

O desafio consistiu num convite aberto ao público em geral para atribuir *tags* e descrever dois conjuntos de aproximadamente 3000 fotos históricas.

O projeto foi bem acolhido no seio da comunidade, tendo registado enorme adesão. O pedido de ajuda público associado à possibilidade de aceder e interagir com as coleções, gerou um movimento massivo e crescente, muito próprio das comunidades da Web 2.0.

[3] http://www.flickr.com/photos/library_of_congress
[4] http://www.steve.museum/

No curto espaço de uma semana a adesão à iniciativa foi enorme e os resultados segundo os organizadores revelaram-se francamente úteis e informativos (Springer, et al., 2008).

Outro exemplo é o projeto Steve, que vive da colaboração entre profissionais de museus e outras entidades que acreditam que o *tagging* social pode providenciar novas formas de descrever e aceder a coleções de objetos culturais, para além de promover a interação com os visitantes.

Segundo Trant (2008), aquando da implementação do protótipo do projeto S*teve.museum* foram comparadas as *tags* atribuídas pelos utilizadores com a documentação do museu, verificando-se que na maioria dos casos os termos utilizados pelos profissionais não coincidiam com os escolhidos pelos utilizadores comuns.

O *tagging* social apresenta-se como um suplemento promissor aos registos dos museus, na medida em que utiliza uma terminologia capaz de suportar alguns tipos de pesquisas (ainda que seja necessário confirmar esta hipótese num estudo a larga escala) (Trant, 2008). Tendo em conta que o utilizador faz as suas pesquisas através da linguagem natural, está sob hipótese a possibilidade de minimizar o distanciamento entre estes e os termos utilizados pelos profissionais.

De facto, segundo Dye (2006) "ainda não é certo que a nova "folksonomia*"* substitua a hierarquia tradicional mas agora que todos os utilizadores têm o poder de classificar segundo a sua própria linguagem, a pesquisa nunca mais será a mesma" (p. 38).

Ainda no mesmo artigo, Trant (2008) evidencia que a opinião geral dos profissionais dos museus é de que a atribuição de *tags* pelos utilizadores poderá ser interessante ainda que a sua pertinência necessite de ser validada.

Contudo, teorias de auto normalização afirmam que as *tags "*folksonomicas" irão autorregular-se, que o vocabulário coletivo será mais consistente com o passar do tempo, sem a imposição externa de um controlo. Numa reação aos constrangimentos do vocabulário controlado, o controlo de sinónimos reduz as nuances e sacrifica o significado, que a hierarquia é frequentemente forçada e falsa e que a "poli-hierarquia" é essencial para se entender a multifacetada natureza do significado. Admitindo que nas terminologias "folksonómicas" a falta de precisão é problemática, ela atribui isto ao comportamento do utilizador em vez de à natureza da *folksonomia* em si e prevê que as *tags* se tornarão auto normalizadoras (Trant, 2009).

Na nossa opinião, as preocupações apontadas podem ser desmistificadas se olharmos para a natureza das *tags*. Neste capítulo, descrevemos o enquadramento feito por Huang e Chuang da natureza das tags na perspetiva da teoria Semiótica. Para além disso, refletimos sobre a forma como o *tagging* social pode contribuir para melhorar o *clustering* de documentos considerando o interpretante das *tags*. Finalmente apresentamos os algoritmos de deteção de comunidades que serão utilizados nesta investigação.

2.1. Enquadramento da Natureza das Tags na Perspetiva da Teoria Semiótica

Para analisar a natureza das *tags* vamos utilizar o enquadramento proposto por Huang e Chuang (2009) no qual é feita uma ligação entre o *tagging* e a teoria semiótica, considerando-se o *tagging* como um sistema de signos. Assim cada recurso, conjuntamente com a sua *tag* e a pessoa que a atribuiu, são tidos como pertencentes a um sistema de signos.

Entende-se por semiótica a ciência que estuda as várias formas de linguagem fazendo para isso uso de signos para representar os objetos. Segundo Peirce (1958), um signo "*is something, A, which denotes some fact or object, B, to some interpretant thought, C*" (capítulo 1, para. 346). Desta forma, qualquer coisa pode ser um signo desde que as pessoas o interpretem como tal não precisando de ter uma presença física; pode ser um fenómeno ou apenas um pensamento (A. W. Huang & Chuang, 2009). Os três componentes do conceito de signo são a representação; o objeto e o interpretante, tal como se ilustra na Figura 26.

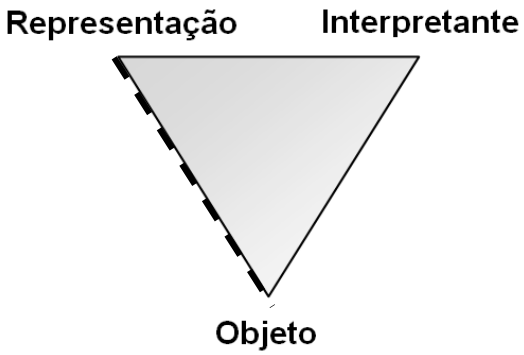


Figura 26: Peirce's *Triadic Sign* adaptado de (A. W. Huang & Chuang, 2009)

- A representação diz respeito à representação do próprio signo, é a forma que o signo assume. No *tagging* são as palavras chave usadas para descrever um recurso no sentido dessa informação ser usada pelo próprio ou por outros.

- O objeto é a entidade a que o signo se aplica. No *tagging*, é o recurso digital a que o *tagger* se refere.
- O *interpretante* de um signo é o sentido ou a interpretação que é feita do signo. No *tagging*, corresponde à interpretação da palavra chave, da descrição ou anotação feita em relação à *tag*, juntamente com os pensamentos do *tagger* sobre o signo.

Neste sentido são identificados como interpretantes desses signos três atores: a comunidade de utilizadores; o autor das *tags* e o designer do sistema.

2.1.1. As Bases da Semiótica de Peirce na sua Fenomenologia

Charles Peirce (1958) denominava a Fenomenologia como a "Doutrina das Categorias", ou "Faneroscopia" a partir do conceito filosófico de *faneron*, um "[…] total coletivo de tudo aquilo que está de qualquer modo presente na mente, sem qualquer consideração se isto corresponde a qualquer coisa real ou não" (capítulo 1, para. 284).

A fenomenologia peirciana distingue 3 categorias de fenómenos (Peirce, 1958): a primeiridade (*firstness),* que está relacionada com o carácter de apresentação do signo; a secundidade (*secondness),* referindo-se ao carácter de representação do signo; e a terceiridade (*thirdness),* relacionada com o poder interpretativo do signo (capítulo 2, para. 243). Tratam-se portanto de estados de consciência em continuidade (Ghizzi, 2009) em que se parte da consciência da qualidade sem qualquer relação, análise ou interpretação, tal como pretende ilustrar o exemplo clássico desta categoria "sentir o vermelho". Neste exemplo refere-se que, quando observamos o vermelho passamos por uma experiência imediata, pura e simples, em que constatamos a cor, não existindo preocupações em conhecer algo mais (sem se querer saber se o vermelho é vermelho de mais alguma coisa). Num segundo momento, segue-se a consciência do outro, que permite reagir, uma experiência direta de compreensão do objeto, ou seja, perceber o objeto como vermelho. Em último lugar, chegamos à consciência sintetizadora, o que possibilita aprender, ou seja interpretar o objeto como vermelho.

A Figura 27 apresenta-nos a cor "vermelho" (*firstness*, na medida em que há uma constatação da existência de uma cor), sendo que esta é a cor iluminada no semáforo (*secondness* – pois há uma verificação da associação da cor vermelho a um determinado objeto), sendo que essa cor, iluminada no semáforo, significa que devemos parar (*thirdness* – interpretação da cor vermelho num determinado contexto). Verifica-se

portanto a existência de uma relação entre as três categorias fenomenológicas de Peirce, podendo ser entendidas como relações entre subconjuntos.

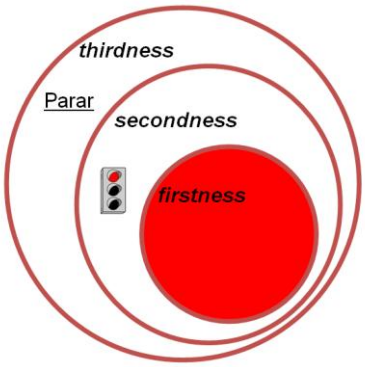


Figura 27: Aplicação das categorias de fenomenologia de Peirce ao vermelho do semáforo.

Segundo Peirce (1958):

> os signos são divididos em três tricotomias: 1) primeiro, de acordo com o signo propriamente dito enquanto mera qualidade, enquanto algo que existe, ou enquanto uma lei geral. 2) em segundo lugar, dependendo se a relação do signo com o objeto que o contém consiste no signo ter sentido por si próprio, ou em alguma relação existencial entre o signo e o objeto, ou na relação do signo com o seu interpretante. 3) em terceiro lugar, dependendo se o interpretante o representa como um signo de possibilidade, como um signo de facto ou como um signo de razão (capítulo 2, para. 243).

Portanto, tendo em conta a tricotomia representação, objeto e interpretante, Peirce, designa para cada um destes, três novos níveis em termos de *Firstness*, *Secondness*, e *Thirdness* (Peirce, 1958). Na Tabela 10 estão apresentados os 9 elementos da tipologia dos signos. Tendo em conta a representação do signo, este pode ser designado por *qualisign*; *signsign* ou *legisign.* Em segundo lugar, o objeto do signo tem três divisões possíveis, *I*con; *Index* ou *Symbol*. Por último, em relação ao interpretante adotou a terminologia, *Rheme*; *Dicisign* e *Dicent sign*.

Tabela 10: Os 9 elementos da tipologia de signos de Peirce.

Categoria/tricotomia	Representação	Objeto	Interpretante
Primeiridade	***Qualisign*** Mera qualidade	***Icon*** Objeto que contém o signo tem sentido em si próprio	***Rheme*** Signo de possibilidade
Secundidade	***Sinsign*** Algo que existe	***Índex*** Relação existencial entre o signo e o objeto	***Dicisign*** Signo de facto
Terceiridade	***Legisign*** Lei geral, uma lei que é um signo	***Symbol*** Relação entre o símbolo e o seu interpretante	***Dicent Sign*** Signo de razão

Huang e Chuang (2009), baseando-se nos três atores do *tagging*, colocaram a hipótese de que o *tagging* social trata as *tags* como signos da comunicação. Sendo assim, adaptaram a terminologia utilizada por Peirce ao *tagging* social combinando-a com a metodologia utilizada por Morris que propõe três dimensões semióticas para os signos, nomeadamente as dimensões sintática (relações formais entre signos), semântica (relações de um signo com o seu objeto) e pragmática (a relação dos signos com os interpretantes), tal como se ilustra na Tabela 11.

Tabela 11: Divisões do fenómeno do *tagging* social segundo Huang e Chuang (2009).

Characters of Triadic Relations	Representation	Sign Relation to Object	Sign Relation to Interpretant		
			Interpretant	Interpreter	Interpretation
1ness Possibility	Mark	Icon	Open	User Community	Community of Interest
2ness Existence	Token	Index	Informational	Tag Writer	Personal Preference
3ness Laws, Rules	Type	Symbol	Formal	System Designer	Semiotic Engineering
	HOW (syntactics)	*WHAT* (semantics)	*WHO & WHY* (pragmatics)		

2.1.2. Adaptação das 10 classes principais de signos segundo Peirce aos diferentes tipos de *Tagging* Social

Das 27 classes de signos possíveis (para cada tricotomia existem 3 hipóteses, $3 \times 3 \times 3 = 27$), apenas 10 classes de signos respeitam a regra da precisão categorial (Peirce, 1958).

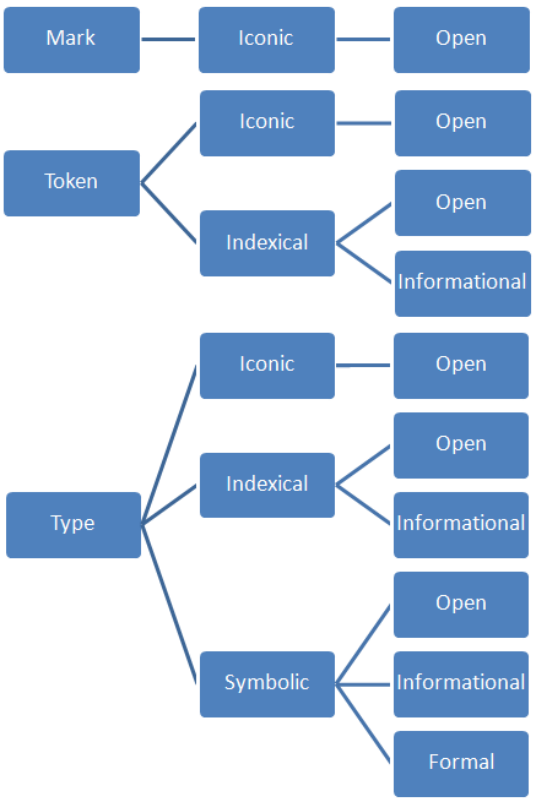


Figura 28: Formação dos dez classes de signos para a classificação do *tagging* social.

Na Figura 29 podemos ver o esquema elaborado por Huang e Chuang (2009) das 10 classes de signos e que tem por base o diagrama de Peirce.

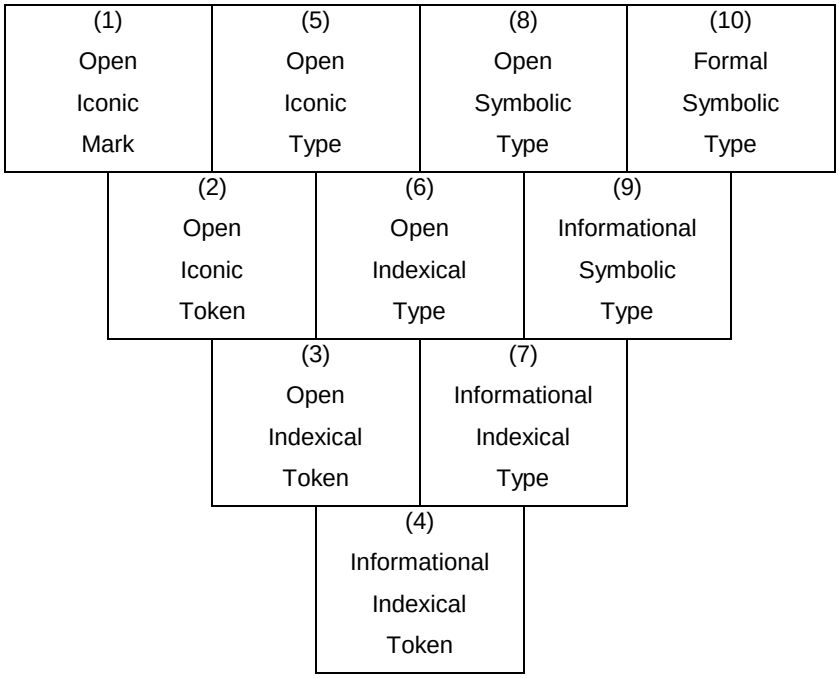


Figura 29: dez classes de signos de Peirce adaptadas por Huang e Chuang (A. W. Huang & Chuang, 2009).

a. Signo (1) – [*Open-Iconic-mark*]

Na primeira classe de signos apresentada por Peirce (1958) este afirma que:

> um Qualisign [por exemplo, um sentimento de "vermelho"] é qualidade desde que seja um signo. Na medida em que uma qualidade é tudo aquilo que tem em si, uma qualidade só pode existir num objeto em virtude de ambos partilharem uma semelhança. Neste sentido um Qualisign é necessariamente um Icon. Para além disso, no sentido em que uma qualidade é uma mera possibilidade lógica, só pode ser interpretada como um signo da essência, ou seja um Rheme (capítulo 2, para. 254).

No *tagging* social, Huang e Chuang, apresentam como exemplos termos como *folksonomy* ou *tagsonomy* que podem ser entendidos como signos uma vez que transmitem sentimentos sobre os conceitos envolvidos no *tagging* (A. W. Huang & Chuang, 2009), o que por sua vez permite estabelecer relações entre taxonomia e o *tagging*. Na Tabela 12, apresentam-se as respostas às questões: Como? O quê? Quem e Porquê?

Tabela 12: Signo (1) na relação triádica (Representante, Objeto, Interpretante) e categorias fenomenológicas.

Categorias	Representação	Relação do signo com o objeto	Relação do signo com o interpretante		
			Interpretante	Intérprete	Interpretação
Primeiridade	***Mark*** Como é um sentimento nenhuma representação determinada o deve representar	***Icon*** Semelhança com outros objetos de *tagging*	***Open***	Comunidade de utilizadores	Comunidade de interesses
	Como? (Sintática)	O quê? (Semântica)	Quem e porquê? (Pragmática)		

b. Signo (2) – [*Open-Iconic-Token*]

Segundo Peirce (1958):

> um Iconic Sinsign [por exemplo um diagrama individual] é qualquer objeto da experiência desde que alguma qualidade sua faça com que ele determine a ideia de um objeto. Sendo um Icon, e portanto um signo por pura semelhança, de tudo quanto possa ser igual, só pode ser interpretado como um signo de essência, ou seja um Rheme. Este signo conterá um Qualisign (capítulo 2, para. 255).

Portanto, é a existência real do signo (1) (A. W. Huang & Chuang, 2009).

No contexto do *tagging* social as *tag clouds* demostram a semelhança dos caracteres e indicam o significado das *tags*. A *cloud* entendida como um *Token* (símbolo) incluí várias cópias de uma *tag* individual (A. W. Huang & Chuang, 2009). Na Figura 30 apresentamos a *tag cloud* das *tags* mais populares de todos os tempos do site *Flickr*. O tamanho da fonte de cada *tag* é proporcional à sua popularidade, fornecendo um sumário visual dos conteúdos.

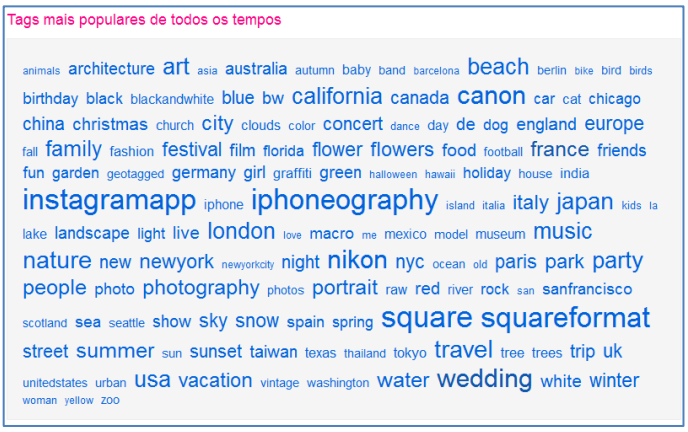


Figura 30: *Tag cloud* das *tags* mais populares do *Flickr*.

Tal como se pode observar na Tabela 13, o representante é de secundidade mas o objeto e o interpretante continuam a ser de primeiridade.

Tabela 13: Signo (2) na relação triádica (Representante, Objeto, Interpretante) e categorias fenomenológicas.

Categorias	Representação	Relação do signo com o objeto	Relação do signo com o interpretante		
			Interpretante	Intérprete	Interpretação
Primeiridade		**Icon** Semelhança com outros objetos de *tagging*	***Open***	Comunidade de utilizadores	Comunidade de interesses
Secundidade	***Token*** *Cloud*				
	Como? (Sintática)	O quê? (Semântica)	Quem e porquê? (pragmática)		

c. Signo (3) – [*Open-Indexical-Token*]

De acordo com Peirce (1958):

> Um Rhematic Indexical Sinsign [por exemplo, um grito espontâneo] é qualquer objeto da experiência direta na medida em que dirige a atenção a um Objeto [semiótico] pelo qual a sua presença é causada. Envolve necessariamente um Iconic Sinsign (signo 2), seja de que tipo for, na medida em que dirige a atenção do intérprete ao próprio objeto indicado (capítulo 2, para. 256).

Por exemplo, uma gargalhada espontânea é um signo de alegria, afetado pelo seu objeto (por exemplo, uma anedota) não fornecendo contudo informações sobre o dito objeto (ou seja, não fornece qualquer informação sobre o conteúdo da dita anedota).

Do ponto de vista do *tagging* social, as listas de *tags* são apresentadas nesta classe como exemplo. Assim, a representação deste signo baseia-se na numeração visual de uma lista de *tags*. Como este signo é um objeto *indexical* representado num *token* (símbolo), significa que existe uma relação existencial entre o objeto a quem foi atribuída a *tag* e a representação da *tag* (A. W. Huang & Chuang, 2009).

Huang e Chuang (2009) referem ainda que as listas de *tags* têm vindo a ser utilizadas para fazer *clustering* de *tags* através da deteção de padrões ou similaridades. Na Figura 31 podemos ver as fotografias que aparecem com a *tag france*. Se optarmos por fazer o *cluster*, aparecem quatro *clusters* e as respetivas *tags* relacionadas (Figura 32).

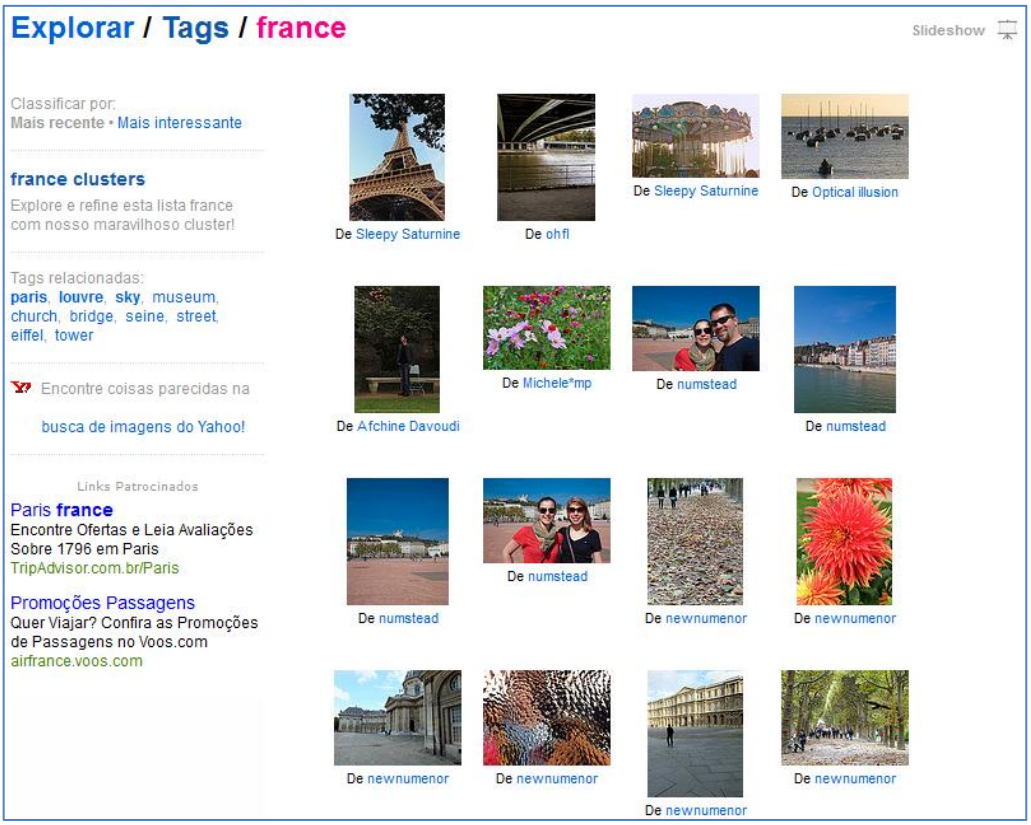


Figura 31: *Tag france* antes do *clustering*.

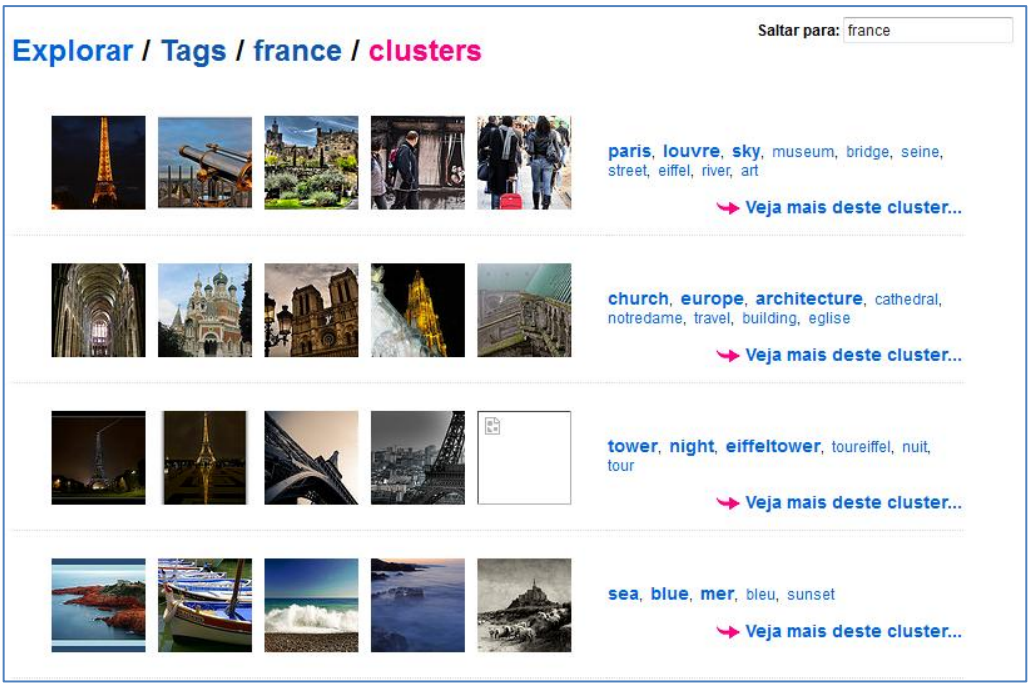


Figura 32: *Tag france* depois de realizado o *clustering*.

Na Tabela 14, observa-se que o representante e objeto são de secundidade ao passo que o interpretante continua a ser de primeiridade.

Tabela 14: Signo (3) na relação triádica (Representante, Objeto, Interpretante) e categorias fenomenológicas.

Categorias	Representação	Relação do signo com o objeto	Relação do signo com o interpretante		
			Interpretante	Intérprete	Interpretação
Primeiridade			**Open**	Comunidade de utiizadores	Comunidade de interesses
Secundidade	***Token*** *Cloud*	***Index*** Lista de *tags*			
	Como? (Sintática)	O quê? (Semântica)	Quem e porquê? (pragmática)		

d. Signo (4) – [*Informational – Indexical - Token*]

Para Peirce (1958) um *Dicent Sinsign* (por exemplo, um catavento) é qualquer objeto resultante da experiência direta, na medida que é um signo, e, como tal, dá informação relativa ao referido Objeto. Tal signo deve envolver um *Iconic Sinsign* (signo 2) para apresentar a informação e um *Rhematic Indexical Sinsign* (signo 3) para indicar o objeto a que a informação se refere. O modo de combinação, ou sintático, destes dois signos tem também de ser significante (capítulo 2, para. 257).

No contexto do *tagging* social, Huang e Chuang (2009) referem que a diferença entre o signo (4) e o signo (3) é que o signo (4) depende principalmente da interpretação das pessoas que atribuem as *tags*, em vez de depender da interpretação dos utilizadores da comunidade. Utilizam a *personomy* para ilustrar este signo no contexto do *tagging* social. Sendo que *personomy* se refere à coleção de *tags* individuais em diferentes sistemas web. Para representar a informação pode ser usada uma *tag cloud* e como objeto *indexical* pode ser usada uma lista de *tags*.

Um exemplo de *personomy* pode ser encontrado em *Tagsahoy*, que tal como sugere a Figura 33, permite pesquisar as *tags* atribuídas pelo utilizador nos seguintes sistemas *LibraryThing*, *Del.icio.us*, *Flickr*, *Gmail*, *Squirl* e *Connotea* listando-as num único sitio[5].

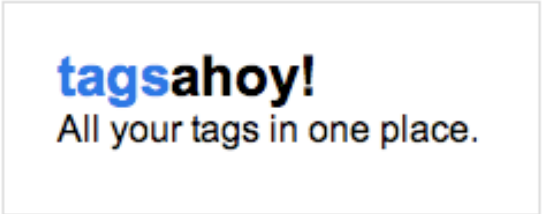


Figura 33: *Tagsahoy*, exemplo de *personomy*

[5] http://www.librarything.com/blogs/librarything/2007/06/tagsahoy/

Neste caso, tal como se pode ver na Tabela 15, o representante, o objeto e o interpretante são de secundidade.

Tabela 15: Signo (4) na relação triádica (Representante, Objeto, Interpretante) e categorias fenomenológicas.

Categorias	Representação	Relação do signo com o objeto	Relação do signo com o interpretante		
			Interpretante	Intérprete	Interpretação
Secundidade	***Token*** *Cloud*	***Index*** Lista de *tags*	***Informational***	Autor da *tag*	Preferências pessoais
	Como? (Sintática)	O quê? (Semântica)	Quem e porquê? (pragmática)		

e. Signo (5) – [*Open – Iconic – Type*]

Um *Iconic Legisign* (por exemplo, um diagrama à parte da sua individualidade factual) é uma lei geral, desde que exija que cada uma das suas características personifique uma qualidade bem definida, capaz de evocar a ideia do seu respetivo objeto. O seu modo ser é o de governar réplicas simples, sendo cada uma delas um *Iconic Sinsign* (signo 2) de um tipo particular (Peirce, 1958, capítulo 2, para. 58).

O signo (5) é muito parecido com o signo (2) mas o signo (5) pode levar a um novo entendimento de possíveis regras. A *tag cloud* associada a uma *tag* específica pode ser um exemplo deste signo. Na medida em que resulta da associação de outras *tags* com uma determinada *tag*. Esta *tag* cloud de uma *tag* específica através da sua representação cria um possível entendimento dessa *tag* por meio de um diagrama icônico (A. W. Huang & Chuang, 2009).

Na Tabela 16, verifica-se que o representante é de terceiridade enquanto que o objeto e o interpretante são de primeiridade.

Tabela 16: Signo (5) na relação triádica (Representante, Objeto, Interpretante) e categorias fenomenológicas.

Categorias	Representação	Relação do signo com o objeto	Relação do signo com o interpretante		
			Interpretante	Intérprete	Interpretação
Primeiridade		***Icon*** Semelhança com outros objetos de *tagging*	***Open***	Comunidade de utilizadores	Comunidade de interesses
Terceiridade	***Type*** *Tag Cloud* de uma *tag*				
	Como? (Sintática)	O quê? (Semântica)	Quem e porquê? (pragmática)		

f. Signo (6) – [*Open-Indexical-Type*]

Um *Rhematic Indexical Legisign* (por exemplo um pronome demonstrativo) é uma lei geral, qualquer que seja a forma como foi estabelecida, que requer que cada um dos seus casos seja efetivamente afetado pelo seu objeto de maneira a que chama a atenção para esse objeto. Cada uma das suas réplicas será o signo (3) de um modo peculiar (Peirce, 1958, capítulo2, para. 259).

No *tagging* social, o signo (6) requer que cada uma das suas instâncias seja grandemente influenciada pelo seu objeto "indexical" de forma a que a atenção da comunidade seja atraída para o objeto (A. W. Huang & Chuang, 2009). Por exemplo, a sigla WOW pode remeter para o jogo online *World* of *Warcraft* e ser utilizada pelos utilizadores de uma comunidade como forma de comunicação. Assim, no contexto específico de utilizadores do jogo online *World of Warcraft*, a *tag* WOW surge como uma regra geral utilizada pelos jogadores para chamar a atenção para o dito jogo.

Tal como se observa na Tabela 17, o representante é de terceiridade, o objeto é de secundidade e o interpretante é de primeiridade.

Tabela 17: Signo (6) na relação triádica (Representante, Objeto, Interpretante) e categorias fenomenológicas.

Categorias	Representação	Relação do signo com o objeto	Relação do signo com o interpretante		
			Interpretante	Intérprete	Interpretação
Primeiridade			**Open**	Comunidade de utilizadores	Comunidade de interesses
Secundidade		***Index*** Chama a atenção para os objetos a que foi atribuída uma *tag* criada pela comunidade			
Terceiridade	***Type*** Criação de novo vocabulário				
	Como? (Sintática)	O quê? (Semântica)	Quem e porquê? (pragmática)		

g. Signo (7) – [*Informational - Indexical - Type*]

O signo (7) (por exemplo, um pregão: Olha o peixe fresquinho) é uma lei geral, qualquer que seja a forma como foi estabelecida, que requer que cada um dos seus casos seja

efetivamente afetado pelo seu objeto fornecendo informações sobre o objeto. Deve conter um signo (5) para apresentar a informação e um signo (6) para designar o objeto da referida informação. Cada réplica deste signo será um signo (4) de um modo peculiar (Peirce, 1958, capítulo 2, para. 260).

No contexto do *tagging* social, o signo (7) distingue-se do signo (4) na medida em que o signo (7), as *tags* são atribuídas por autores individuais e são usadas para categorizar os objetos preferidos num dado sistema (A. W. Huang & Chuang, 2009), implicando que o ator do *tagging* centre a sua atenção num dado objeto, retirando informação específica sobre o objeto, enquanto que no signo (4) existe apenas a listagem de uma coleção de *tags* atribuídas ao longo de vários sistemas, sem existir a preocupação de as categorizar.

h. Signo (8) – [*Open – Symbolic – Type*]

Este signo está ligado ao seu objeto por qualquer ligação de ideias gerais. O signo lida com uma regra aberta formal que gera uma explicação geral do signo. O interpretante do signo (8) por vezes representa o signo como um signo (6) e outras vezes como um signo (5) porque partilham algumas características. A réplica do signo (8) corresponde a um determinado tipo de signo (3). Note-se que o signo (8) é diferente do signo (6) na medida em que as suas respetivas réplicas no signo (3) não podem ser do mesmo tipo (Peirce, 1958; A. W. Huang & Chuang, 2009).

Numa perspetiva semiótica verifica-se que a interpretação de um signo (8) está associada ao recurso a que este se refere. Contudo, a informação contida numa *tag* usando palavras comuns (signo (8)) pode não ser tão específica como a que está presente nas *tags* criadas por comunidades (signo (6)). Isto acontece porque o signo (8) partilha a natureza dos signos (5) e (6). O signo (8) também difere do signo (6) na medida em que a suas respetivas réplicas no signo (3) não podem ser do mesmo tipo. A regra aberta formal existente no signo (8) é visível na atribuição de *tags* que são palavras comuns (A. W. Huang & Chuang, 2009), palavras da linguagem natural.

i. Signo (9) – [*Informational – Symbolic – Type*]

O signo (9) é um signo ligado ao seu objeto através de uma associação de ideias gerais e que atua como o signo (8), tirando o facto do interpretante previsto dever estar ligado ao objeto indicado. Apesar do signo (9) partilhar uma parte da natureza do signo (7) o seu objeto simbólico é diferente na medida que a sua representação é vista essencialmente como uma declaração de facto. A réplica do signo (9) é o signo (4) de um determinado tipo (Peirce, 1958; A. H. Huang & Chuang, 2009).

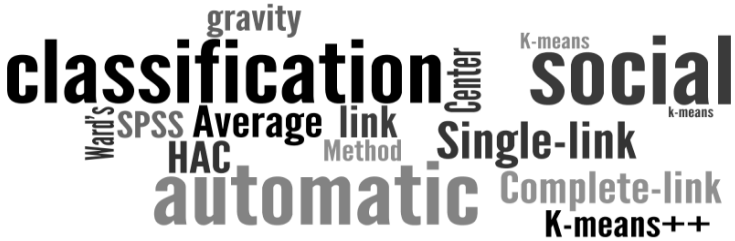


Figura 34: *Tag cloud* das *tags* atribuídas a uma amostra dos recursos utilizados na revisão de literatura para a escrita desta tese.

O signo (9) é normalmente visto como *tagging* pessoal mas pode também ser observado nas formas linguísticas das *tags* de representações não nominais. Estas formas são vistas como suplementos de categorias que vão buscar os seu significado às descrições das categorias. O *tagging* pessoal partilha algumas das características do signo (7); contudo o objeto simbólico do signo (9) torna a sua representação maioritariamente uma afirmação de facto. Se o signo (9) é visto como *tagging* pessoal, a sua réplica é também uma *personomy* de um tipo específico (signo(4)) (A. W. Huang & Chuang, 2009). Por exemplo, na Figura 34, podemos ver uma *tag cloud* das *tags* atribuídas a parte dos recursos utilizados para fazer a revisão de literatura desta tese. A figura conta parte da história desta tese, podemos ver que aqui são abordadas a Classificação Social e algoritmos de *clustering*.

j. Signo (10) - [*Formal – Symbolic – Type*]

Segundo Peirce (1958):

> Um argumento é um signo cujo interpretante representa o seu objeto como sendo um signo escondido através de uma lei, nomeadamente, segundo a qual a passagem por tais premissas para formular determinadas conclusões tende a ser verdadeira. Manifestamente, então, o seu objeto deve ser geral; isto é, o argumento deve ser um *Symbol* (capítulo 2, para. 263).

A réplica do signo (10) é o signo (4).

O signo (10) é, de acordo com Huang e Chuang (2009), o signo mais complexo no contexto do *tagging* social. Neste caso, sendo o interpretante o designer do sistema, este tem como preocupação os métodos de classificação social, de categorização colaborativa, indexação em grupo e a "etnoclassificação", pelo que procura que as suas interpretações sejam o resultado de regras formais generalizadas a partir de interpretações de comunidades de utilizadores, autores de tags e dos próprios designers dos sistemas.

Observe-se que no site *Delicious*, quando selecionamos um marcador, o sistema apresenta por defeito uma proposta de *tags* geradas com base na *tag cloud* do utilizador e da comunidade do *Delicious* (Figura 35).

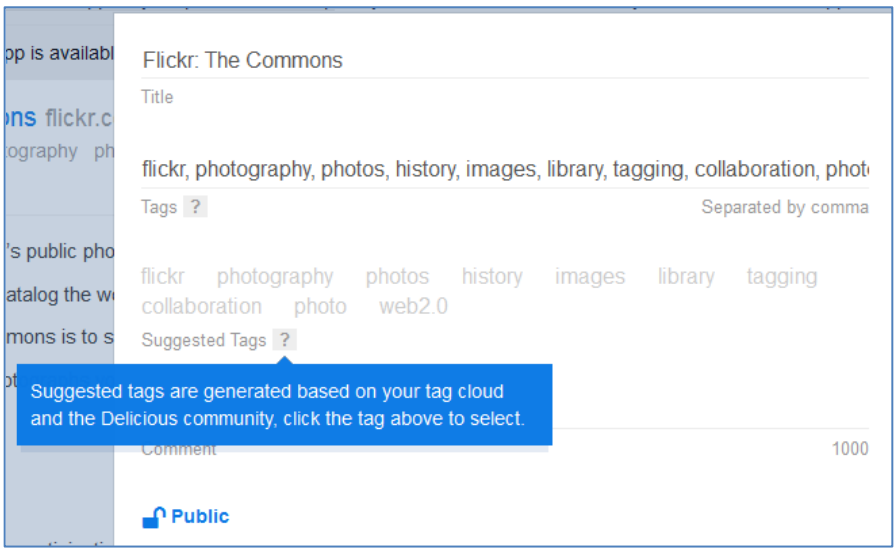


Figura 35: *Tags* sugeridas pelo sistema *Delicious*.

Na Figura 36, podemos ainda encontrar recomendações do sistema para atribuição de *tags*, tais como a utilização de letras minúsculas, evitar os espaços entre palavras, reutilizar *tags* próprias e da comunidade e a utilização do ponto de exclamação para identificar *tags* pessoais que dificilmente seriam úteis para a comunidade.

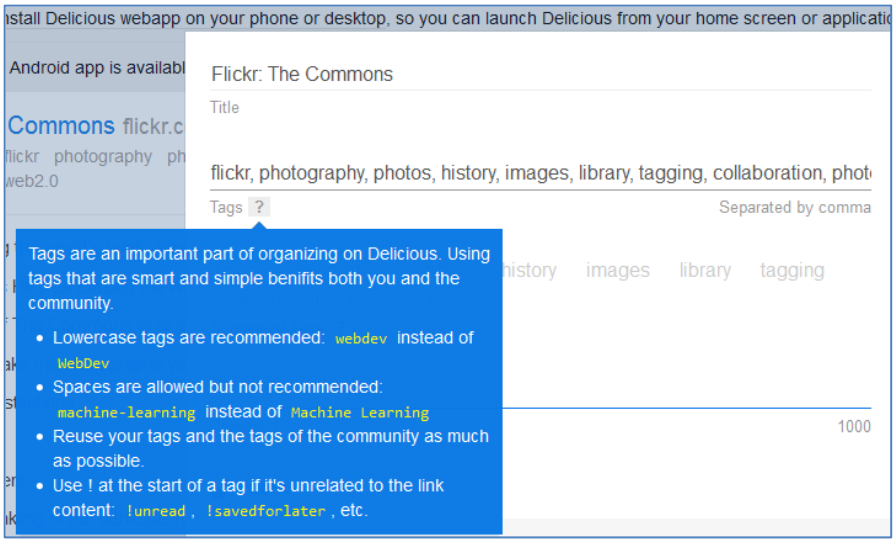


Figura 36: Recomendações do sistemas para a atribuição de *tags*.

Em síntese, Huang e Chuang (2009), no que respeita à questão de como é que o *tagging* social se liga às comunicações online (Tabela 18), identificam que os signos de (5) a (10) providenciam regras gerais de comunicação online; que os signos de (2) a (4)

representam a atual existência do *tagging* em relação à comunicação social e por fim o signo (1) que traduz o possível conceito de *tagging* social no processo de comunicação online.

Tabela 18: Como é que o *Tagging* Social se liga às comunicações online?

Signos	Como?
(5) Tag Cloud of Tag; (8) Common Tag; (10) Tag in System; (6) Community Indexical Tag; (9) Personal Tag; (7) Personal Indexical Tag	Providenciam regras gerais de comunicação online.
(2) Tag Cloud; (3) Tag List /Clustering; (4) Personomy	Representam a atual existência do tagging relativamente à comunicação social.
(1) Tagging Folksonomy	Possível conceito do tagging Social no processo de comunicação online.

Por outro lado, no que respeita à questão a que objetos as *tags* se referem, como mostra a Tabela 19, temos signos onde se determinam as semelhanças características dos objetos a que as *tags* se referem; signos que relacionam a existência dos objetos com os usos da comunicação e ainda signos que são usados para leis ou regras.

Por fim, no que respeita, à questão quem são os intérpretes dos signos e o porquê dessa interpretação, temos 3 níveis de interpretantes. Assim, como se pode ver na Tabela 20,

temos signos interpretados pelos utilizadores da comunidade em função de interesses passados e presentes da comunidade; signos que são interpretados pelos autores das *tags* de acordo com as suas preferências pessoais e por fim os signos que são interpretados pelos designers de sistema através da criação de regras e sugestões para a escrita de *tags*.

Tabela 19: A que objetos as *tags* se referem?

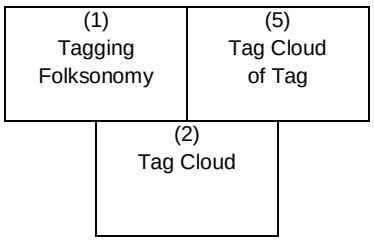
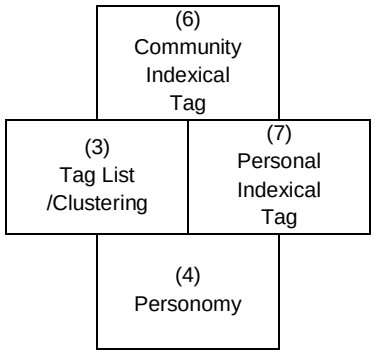
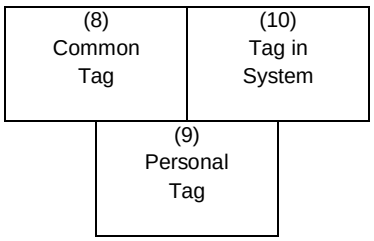
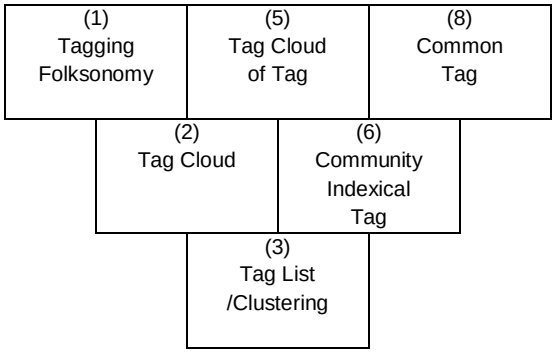
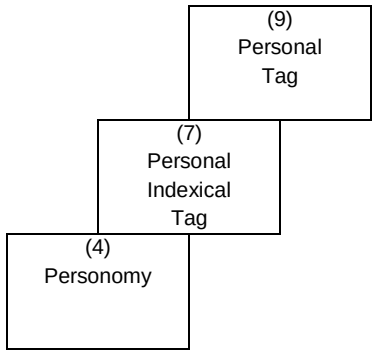
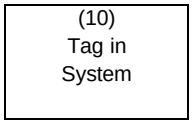
Signos	A que objetos as *tags* se referem?
	Os 3 *icons* para determinar as semelhanças características dos objetos a que as *tags* se referem.
	Os índexes relacionam-se com a existência dos objetos indicados e com os usos da comunicação.
	Os 3 símbolos são usados para certas leis ou regras.

Tabela 20: Quem são os intérpretes e porquê?

Signos	Quem e porquê?
	Utilizadores de comunidades e agem em função dos interesses da comunidade.
	Autores das *tags* criadas em função de preferências pessoais.
	Designers de sistema. Estes desenvolvem regras para permitir o raciocínio lógico e a meta-comunicação.

2.2. Como é que o *Tagging* pode Contribuir para Melhorar o *Clustering* de Documentos?

A procura de um consenso social sobre a forma como os recursos de um determinado sistema devem estar organizados é no mínimo controversa, uma vez que a podemos fazer de variadas formas e todas elas serem válidas. Numa perspetiva da teoria semiótica, podemos considerar que estamos dependentes do interpretante para perceber como os recursos deverão estar organizados.

Enquanto utilizador de uma comunidade, a interpretação das tags pode ser feita de acordo com os interesses da comunidade mas do ponto de vista do autor da tag, a

interpretação poderá basear-se nas suas preferências pessoais. Desta forma, vamos analisar as relações entre o signo (3) (*tag list*/*clustering*), o signo (6) (*community Indexical Tag*) e o signo (8) (*Common Tag*) cujo interpretante é a comunidade de utilizadores, e a relação entre os signo (4) (*personomy*), o signo (7) (*Personal Indexical Tag*) e o signo (9) (*Personal Tag*) cujo interpretante é o autor da *tag*.

Nenhum sistema de classificação funciona para todos, em todas as culturas e ao mesmo tempo. Da mesma forma, do ponto de vista das comunicações online, a forma como estão organizados os recursos será diferente para os três tipos de atores do *tagging* e mesmo para os mesmos atores de *tagging*, as mesmas *tags* podem ser interpretadas de maneira diferente.

No contexto desta investigação procuramos perceber de que forma o *tagging* social pode contribuir para melhorar a qualidade do *clustering* de documentos. Portanto é necessário percebermos como os signos de *tagging* aparecem na atividade de *tagging*.

2.2.1. Interpretação Segundo a Comunidade de Utilizadores

Partindo de uma lista de *tags*, vamos analisar de que forma se relacionam os signos (3), (6) e (8), uma vez que os signos (6) e (8) têm como réplica o signo (3). No signo (3), partindo das listas de *tags*, são utilizadas técnicas que providenciam métodos para *clustering*. Já o signo (6) apresenta *tags* com formas que não aparecem no vocabulário (por exemplo uma sigla), sendo *tags* mais específicas do que, por exemplo, aquelas que são tratadas no signo (8), as palavras da linguagem natural (portanto mais gerais e consequentemente mais populares).

Vamos analisar uma lista de *tags* numa tentativa de compreensão de como podemos interpretar o agrupamento de documentos sob a perspetiva da comunidade de utilizadores.

Para isso, consideramos uma amostra de 5000 documentos da *Wikipedia* retirados do repositório disponibilizado online pelo NLP *Group*[6] onde verificamos que em média 57% das *tags* foram atribuídas apenas por um *tagger* (Gráfico 1). Para além disso, verifica-se que a média e a mediana das *tags* atribuídas por 2 *taggers* é de aproximadamente 10%. À medida que o número de *taggers* que atribui a mesma *tag* aumenta, a percentagem da média e da mediana tende para valores próximos os 0%.

[6] http://nlp.uned.es/social-tagging/wiki10+/

Para além disso, algumas das *tags* que só são atribuídas uma vez estão relacionadas com tags já atribuídas por mais do que um *tagger*. Por exemplo, *muusika* (1 voto) e *music* (15 votos); *type_font_designers* (1 voto), *type* (2), *font* (14 votos) *designer* (4 votos). Para uma amostra de 25 recursos, verificamos que, em média, aproximadamente 15% das *tags* (atribuídas apenas por um utilizador) já estão representadas por *tags* atribuídas por mais de um utilizador.

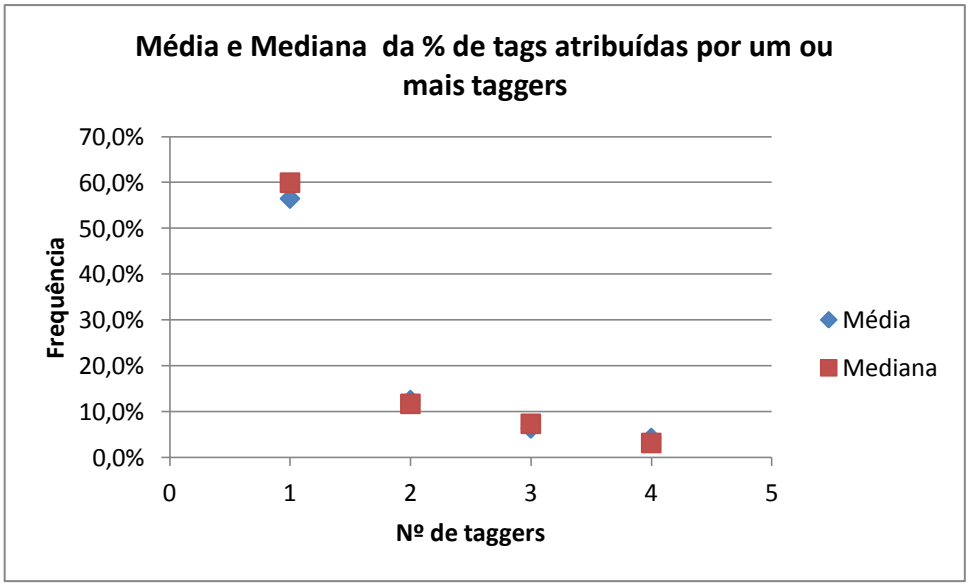


Gráfico 1: Média e Mediana da % de *tags* atribuídas por um ou mais *taggers*.

Para além disso, verificamos que a *tag* atribuída por mais *taggers* corresponde em média a 28,6% do total de todas as ocorrências de todas as *tags* de cada recurso, que em média são 23,2 *tags* por recurso.

Podemos ainda constatar que entre as *tags* atribuídas a cada recurso, temos *tags* mais específicas (signo (6)) e *tags* mais gerais como é o casos das *tags* presentes no signo (8).

Ora, a organização dos documentos em *clusters* usando todas as *tags*, fará sempre sentido? Sobretudo quando verificamos que mais de metade das *tags* foi atribuída por apenas 1 utilizador? Acreditamos que quantos mais utilizadores concordarem que determinada *tag* caracteriza o recurso maior será o consenso. Aliás, quando usamos *keywords* para um artigo, usamos geralmente entre 4 e 5, não 20. Contudo, não pretendemos impor um número mínimo de ocorrências que deve ter uma *tag* para poder ser utilizada (deixamos isso ao critério do utilizador pois temos de admitir que pode ser essencial utilizar todas as *tags*).

a. Grau de Consenso

Entendemos que ao utilizador de um sistema poderá ser dada a possibilidade de escolher um parâmetro para definir o grau de consenso (gc, p) onde $p \in]0,1]$ que controlará a formação dos *clusters*. Por exemplo, se escolher $gc = 3$ e $p = 0{,}25$, significa que para o *Clustering* selecionam-se as *tags* com 3 ou mais ocorrências mas destas só serão utilizadas 25% (as que têm mais ocorrências).

Desta forma, um utilizador enquanto membro de uma comunidade pode interpretar o todo de acordo com a sensibilidade que tem da respetiva comunidade, reorganizando dinamicamente os recursos.

2.2.2. Interpretação Segundo os Autores das *Tags*

Os signos (4), (7) e (9) estão relacionados uma vez que os signos (7) e (9) têm como réplica o signo (4). No signo (4), está normalmente incorporado nos sistemas que permitem disponibilizar num único site as *tags* individuais atribuídas ao longo de vários sistemas. Parte de uma lista de *tags* e tem o conceito de *personomy* associado. Já no signo (7) as *tags* atribuídas são utilizadas para categorizar os recursos preferidos num dado sistema, (envolve o signo (6) uma vez que é necessário indicar o assunto e é apresentado através do signo (5)). O signo (9) atua como o signo (8) com a diferença de que o interpretante tem de estar ligado ao recurso.

Portanto, o *Clustering* tendo por base a presença destes signos terá interpretações diferentes, podendo os mesmos recursos serem agrupados de formas diferentes por diferentes autores de *tags*.

2.3. Deteção de Comunidades

O campo de Deteção de Comunidades é relativamente recente e assume grande importância em áreas como a Ciência dos Computadores, Biologia e Sociologia, onde os sistemas são frequentemente representados por grafos (Fortunato & Castellano, 2009).

As redes complexas, neste caso as redes de *tags*, são representadas através de grafos e procede-se à organização dos vértices em comunidades (*clusters*) com muitas arestas unindo vértices da mesma comunidade e relativamente poucas a unir vértices de comunidades diferentes.

De seguida, apresentamos os dois algoritmos utilizados nesta investigação para detetar comunidades numa rede de *tags*. O algoritmo proposto por Girvan e Newman (2002), considerado por Fortunato e Castellano (2009) um marco no campo da Deteção de

Comunidades e o algoritmo Wakita-Tsurumi (Wakita & Tsurumi, 2007) um algoritmo mais eficiente para redes mais complexas.

2.3.1. Girvan e Newman

O algoritmo proposto por Girvan e Newman (2002) começa por retirar as arestas com maior *betweenness centrality*.

Para calcular a *betweenness centrality* calcula-se o número de caminhos mais curtos que passam por uma determinada aresta. Observe-se a Figura 37, onde está representado um grafo com a indicação da *betweenness centrality* de cada aresta. Por exemplo, se considerarmos a aresta DE, temos de contabilizar todos os caminhos mais curtos que passam por DE. Assim, temos os caminhos A-E; A-F; A-G; A-H; B-E; B-F; B-G; B-H; C-E; C-F; C-G; C-H; D-E; D-F; D-G e D-H.

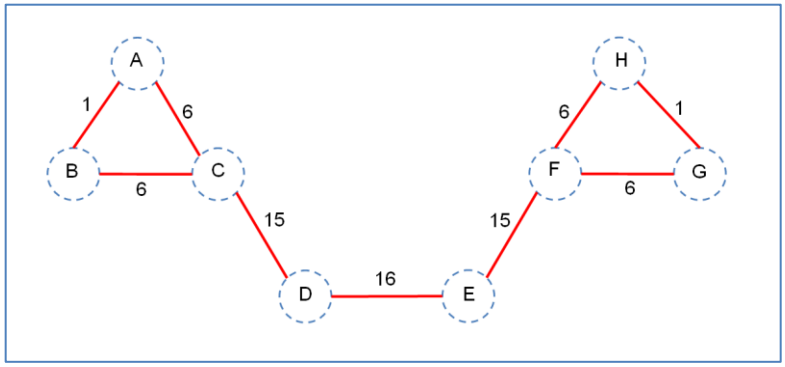


Figura 37: Cálculo da *betweenness centrality* de cada aresta do grafo.

No caso de existir mais do que um caminho mais curto, todos os caminhos são igualmente pesados de modo a que o total dê a unidade. Por exemplo, na Figura 38, para chegar de D a B passando pela aresta EA existem dois caminhos curtos: D-E-C-B e D-E-A-B, logo cada caminho tem um peso de 0,5.

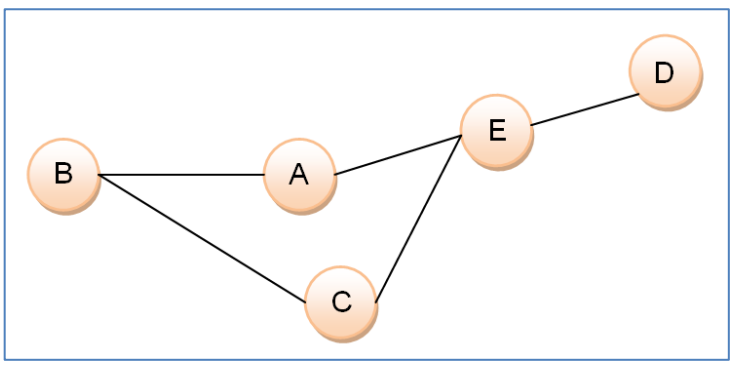


Figura 38: Grafo com mais do que um caminho mais curto entre dois vértices.

Algoritmo

1: Calcula-se a *betweenness centrality* de cada aresta.

2: Remove-se a aresta com maior centralidade (escolha aleatória se ocorrer empates entre arestas).

3: Recalculam-se as centralidades do novo grafo e volta ao passo 2.

Complexidade

A complexidade do primeiro passo do algoritmo é $O(n^2)$. Adicionalmente, estudos experimentais mostram que o recálculo do passo 3 introduz um fator adicional m ao tempo de execução do algoritmo, podendo ser $O(m^2 n)$ ou $O(n^3)$ se o grafo for esparso (Fortunato & Castellano, 2009).

2.3.2. Modularidade

A modularidade é uma função de qualidade sendo a mais popular a de Girvan e Newman (2004), introduzida inicialmente para definir um critério de paragem do algoritmo Girvan e Newman mas que rapidamente se tornou um elemento essencial de muitos métodos de *clustering* (Fortunato & Castellano, 2009). Esta função tem por base a ideia de que não é esperado encontrar uma estrutura de grupos num grafo aleatório, pelo que a possível existência de grupos é revelada através da comparação da densidade das arestas no subgrafo e a densidade que esperamos encontrar no subgrafo no caso dos vértices estarem unidos, independentemente da estrutura da comunidade. A densidade expectável das arestas depende do modelo nulo selecionado (pode ser por exemplo uma cópia do grafo original mantendo algumas das suas propriedades estruturais mas sem a estrutura de comunidade) (Fortunato & Castellano, 2009).

A modularidade é então dada pela Equação 12.

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - P_{ij}\right)\delta\left(C_i, C_j\right) \qquad \text{Equação 12}$$

Onde:

- A é a matriz de adjacência;
- m é o número total de arestas do grafo;
- P_{ij} representa o número esperado de arestas entre os vértices i e j no modelo nulo;
- A função δ assume o valor de um se os vértices i e j estiverem na mesma comunidade $(C_i = C_j)$, e zero se assim não for.

O algoritmo Girvan e Newman é um algoritmo hierárquico e por isso a modularidade é calculada a cada evolução do dendrograma e no final é escolhido o melhor Q.

O primeiro algoritmo aglomerativo proposto para maximizar a modularidade foi implementado por Newman (2004). Neste algoritmo a cada iteração é necessário calcular a variação ΔQ da modularidade obtida a partir da fusão de quaisquer duas comunidades da partição em análise de forma a que possa ser escolhida a melhor fusão. Depois de decidir quais a comunidades a fundir, é necessário fazer uma atualização à matriz e_{ij} expressando a fração das arestas entre os *clusters* i e j da partição que está a ser executada.

Clauset *et al*. (2004) propuseram num artigo posterior uma otimização ao algoritmo proposto por Newman. Os autores consideram que o algoritmo pode ser executado de forma mais eficiente utilizando estruturas de informação para matrizes dispersas. Este método permite analisar as estruturas das comunidades em grafos de grandes dimensões que contenham até 10^6 vértices.

Wakita e Tsurumi (2007) observaram que devido à tendência para a formação de comunidades muito grandes, o algoritmo proposto por Clauset *et al.* (2004) revelou-se ineficiente na medida em que gera dendrogramas pouco equilibrados, levando a que o tempo de execução se aproxime do pior caso. Para melhorar a situação propuseram uma modificação que, a cada passo, procura fazer fusões de comunidades devolvendo o valor mais alto do produto de ΔQ por um fator que denomina rácio de consolidação, resultando em comunidades com tamanhos aproximados (Fortunato & Castellano, 2009). Esta alteração permite analisar sistemas que contenham até 10^7 vértices.

Capítulo 3
Métodos de Integração das *Tags* no *Clustering* de Texto

Neste capítulo propomos dois métodos para integrar as *tags*, partindo do algoritmo de *clustering k-means*. Selecionámos este algoritmo porque é considerado muito eficiente e julgamos que os problemas que apresenta, nomeadamente a seleção do k e da seleção aleatória das sementes podem através da rede de tags ser colmatados. Para além disso, é considerado um dos "*top 10 algorithms* " em *data mining* (Wu, et al., 2007).

O primeiro método permite a integração das *tags* no vetor dos documentos através de um parâmetro chamado *Social Slider* (SS) que permite atribuir diferentes pesos às *tags* em função da sua ocorrência no documento. No sentido de prever o impacto da sua integração será apresentado um modelo teórico de previsão tendo em conta as medidas de similaridade usualmente utilizadas para executar o algoritmo *k-means*.

O segundo modelo de integração é baseado na Deteção de Comunidades numa rede de *tags*, permitindo uma escolha cuidada das sementes. Este novo algoritmo denominado *k-Communities* (k-C) diferencia-se do algoritmo *k-means* não só pela seleção das sementes mas também porque introduz uma nova forma de calcular os centroides em cada iteração.

3.1. Modelo Matemático para Integração das Tags num Vector Space Model

Uma *tag* pode aparecer num documento mais do que uma vez, apenas uma vez ou nunca. Como se ilustra na Figura 39, a *tag* x aparece mais do que uma vez, a *tag* y apenas uma vez e a *tag* z nunca aparece.

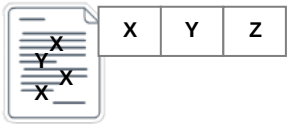


Figura 39: No documento a *tag* x aparece mais do que uma vez, a *tag* y apenas uma vez e a *tag* z nunca aparece.

Intuitivamente, uma *tag* que também aparece no texto parece ser mais relevante do que uma *tag* que só aparece uma vez ou que nunca aparece. Partimos assim para a construção de um modelo matemático que permita atribuir pesos às *tags* tendo em conta a sua ocorrência no documento.

Seja $D = \{d_1, d_2, \dots, d_N\}$ um conjunto de n documentos; $W = \{w_1, w_2, \dots, w_T\}$ e $T = \{t_1, t_2, \dots, t_T\}$ sejam respetivamente o conjunto de termos e de *tags* que podem aparecer nos documentos.

Para integrar no documento a informação proveniente das *tags*, consideram-se dois vetores: o vetor constituído apenas por *tags* (Vt_j) e o vetor do documento (Vd_j). Temos então $Vd_j =< fw_{1j}, fw_{2j}, fw_{3j}, \dots, fw_{Tj} >$ e $Vt_j =< ft_{1j}, ft_{2j}, ft_{3j}, \dots, ft_{Tj} >$, onde fw_{ij} representa a frequência com que o termo p_i ocorre no documento d_j e ft_{ij} representa a frequência com que cada tag t_i foi atribuída no documento d_j.

Como se pode ver na Figura 40, cada vetor de *tags* (Vt_j) é alterado: a cada coordenada é adicionada a frequência da *tag* no documento, o resultado é multiplicado pelo parâmetro *Social Slider* (SS) que está dependente da ocorrência da tag no documento. Depois de atualizado o vetor das *tags*, este é adicionado ao vetor do documento, permitindo assim que o peso atribuído a cada *tag* seja integrado no documento.

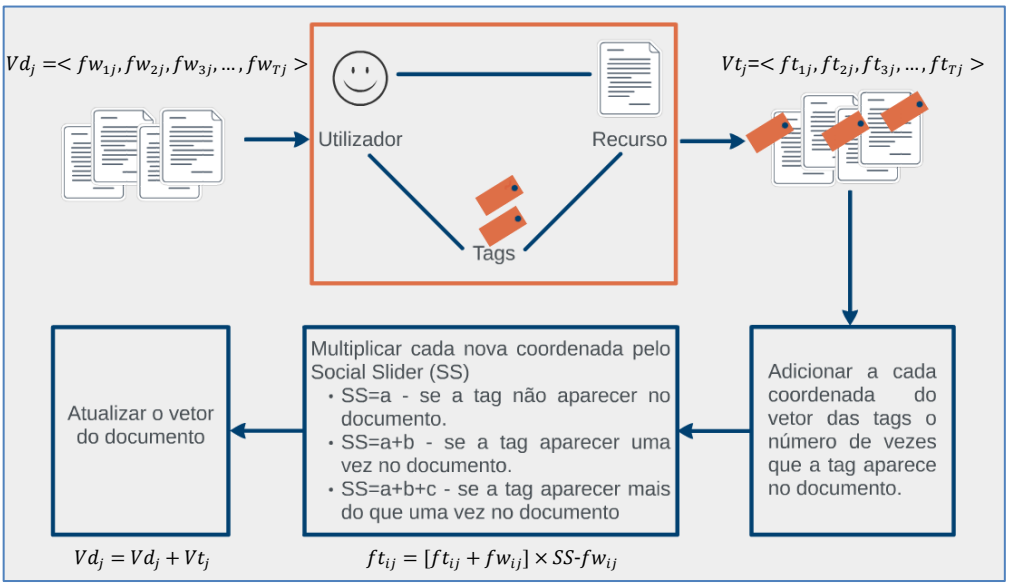


Figura 40: *Tags* no *Vector Space Model*

O valor do parâmetro SS será determinado pelo utilizador tendo em conta a importância que pretende dar à classificação social.

3.2. Modelo Teórico para prever o impacto das tags

O modelo de integração proposto deixa emergir a necessidade de prever o impacto da integração do peso das *tags* no documento, isto é, se os documentos que partilham *tags* ficam mais próximos ou se ficam mais distantes quando não partilham *tags*. Portanto, a previsão deste impacto dependerá certamente da medida de similaridade utilizada.

As medidas de similaridade mais frequentemente utilizadas no algoritmo *k-means* são a distância Euclidiana e a similaridade dos cossenos.

Distância Euclidiana

Utilizado a distância Euclidiana temos que a distância entre dois documentos X e Y é dada pela Equação 13.

$$\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} \qquad \text{Equação 13}$$

Onde $X = (x_1, x_2, \ldots, x_n)$ e $Y = (y_1, y_2, \ldots, y_n)$

Supondo que foi atribuída a mesma *tag* aos documentos X e Y, obtemos:

$$\sqrt{(x_1 - y_1)^2 + ((x_2 + a) \times SS_X - (y_2 + b) \times SS_Y)^2 + \cdots + (x_n - y_n)^2} \qquad \text{Equação 14}$$

Onde a e b correspondem à frequência com que a *tag* foi atribuída ao documento X e Y respetivamente.

SS_X e SS_Y dependem da ocorrência das palavras no conteúdo do documento.

Sejam $a \geq 0$, $b \geq 0$ e $SS > 0$

Se $(x_2 - y_2)^2 = 0$ então $((x_2 + a) \times SS_X - (y_2 + b) \times SS_Y)^2 \geq 0$

Se $(x_2 - y_2)^2 > 0$ então $((x_2 + a) \times SS_X - (y_2 + b) \times SS_Y)^2 \geq 0$

Isto significa que tomando o parâmetro SS superior a zero, só existe uma possibilidade de reduzir a distância entre os documentos, ou seja quando antes da integração o quadrado da diferença das coordenadas é superior a zero e depois da integração das

tags esta diferença fica igual a zero. Portanto, é trivial concluir que o parâmetro SS teria de variar entre 0 e 1 de modo a permitir que a distância entre os documentos diminuísse.

Similaridade dos cossenos

Por outro lado, o cosseno do ângulo formado entre dois vetores pode variar entre 0 e 1 e, respetivamente, o ângulo pode variar entre 90 graus e 0 graus. Uma vez que a cada coordenada do vetor corresponde a frequência com que a palavra ocorre no documento, esta será sempre não negativa.

Portanto, se calcularmos o cosseno do ângulo x entre dois documentos antes de integrarmos as *tags*, obtemos:

$$\cos(x) = \frac{\sum_{i=1}^{n} y_i \times z_i}{\|Y\| \times \|Z\|} \qquad \text{Equação 15}$$

Onde $Y = (y_1, y_2, \dots, y_n)$ e $Z = (z_1, z_2, \dots, z_n)$

Assim, para cada um dos pares de documentos descritos anteriormente, pretende-se analisar a influência da integração das *tags*. Para tal, calcula-se o cosseno do ângulo formado entre os documentos depois da integração das *tags* (cos(a)). Assim, escrevendo o cosseno do novo ângulo através das coordenadas iniciais, é possível fazer variar o valor de SS de modo a permitir visualizar o que acontece, por exemplo, a um par de documentos que apresentava inicialmente cos(x)=0,1 e que, depois da integração das *tags* para um determinado valor de SS, passa a apresentar cos(a)=0,6, indiciando que o peso dado à classificação social permitiu aproximar documentos que inicialmente estavam muito afastados e que eventualmente estariam em *clusters* diferentes.

Para as situações ilustradas nas figuras abaixo serão exibidos três gráficos onde as normas iniciais dos vetores andam próximas de 10, 30 e 100.

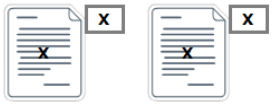
De seguida apresentamos os casos que serão analisados.

Documentos que partilham a mesma *tag* quando esta:

- Aparece nos documentos mais do que uma vez;

- Nunca aparece nos documentos;

- Aparece uma única vez nos documentos.



Segue-se a análise de como pode variar a aproximação entre os documentos que partilham a mesma *tag*, sendo que esta pode aparecer no conteúdo do texto com frequências diferentes, tal como ilustrado na Figura 41.

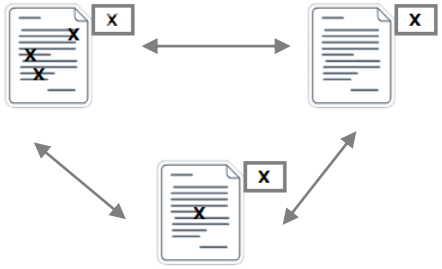


Figura 41: Documentos que partilham a mesma *tag* mas com frequências diferentes no conteúdo do documento.

Para além disso, será analisado como cada um dos tipos de atribuições de *tags* pode influenciar o ângulo que é formado com os restantes documentos onde não coocorrem *tags*, como mostra a Figure 42.

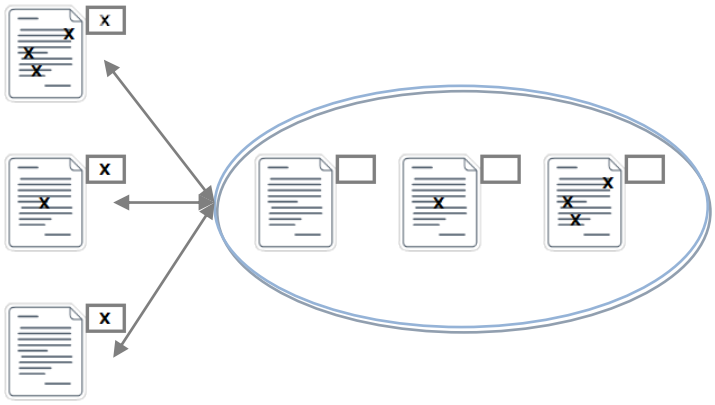


Figure 42: Documentos que não partilham *tags*.

3.2.1. Documentos com a mesma *tag*

Usando o modelo de integração, e sem perda de generalidade, vamos supor que as coordenadas z_j e y_j correspondem à frequência da mesma palavra, coincidente com a *tag* associada aos dois documentos (à *tag* será atribuída frequência 1). Portanto, as coordenadas que têm *tags* associadas são atualizadas permitindo obter, através de uma manipulação algébrica, o cosseno do ângulo formado pelos novos vetores de documentos - cos(a):

$$\cos(a) = \frac{z_1 \times y_1 + \cdots + (z_j + 1) \times SS \times (y_j + 1) \times SS + \cdots + z_n \times y_n}{\sqrt{z_1^2 + \cdots + (z_j + 1)^2 \times SS^2 + \cdots z_n^2} \times \sqrt{y_1^2 + \cdots + (y_j + 1)^2 \times SS^2 + \cdots y_n^2}}$$ Equação 16

$$= \frac{(SS^2 - 1) y_j z_j + SS^2 y_j + SS^2 z_j + SS^2 + z_1 \times y_1 + \cdots + z_j \times y_j + \cdots + z_n \times y_n}{\sqrt{z_1^2 + \cdots + (z_j + 1)^2 \times SS^2 + \cdots z_n^2} \times \sqrt{y_1^2 + \cdots + (y_j + 1)^2 \times SS^2 + \cdots y_n^2}}$$

$$= \frac{(SS^2 - 1) y_j z_j + SS^2 y_j + SS^2 z_j + SS^2 + \sum_{i=1}^{n} z_i \times y_i}{\sqrt{\sum_{i=1}^{n} z_i^2 + (SS^2 - 1) z_j{}^2 + 2SS^2 z_j + SS^2} \times \sqrt{\sum_{i=1}^{n} y_i^2 + (SS^2 - 1) y_j{}^2 + 2SS^2 y_j + SS^2}}$$

$$= \frac{(SS^2 - 1) y_j z_j + SS^2 y_j + SS^2 z_j + SS^2}{\sqrt{\sum_{i=1}^{n} z_i^2 \left(1 + \frac{(SS^2 - 1) z_j{}^2 + 2SS^2 z_j + SS^2}{\sum_{i=1}^{n} z_i^2}\right)} \times \sqrt{\sum_{i=1}^{n} y_i^2 \left(1 + \frac{(SS^2 - 1) y_j{}^2 + 2SS^2 y_j + SS^2}{\sum_{i=1}^{n} y_i^2}\right)}}$$

$$+ \frac{\sum_{i=1}^{n} z_i \times y_i}{\sqrt{\sum_{i=1}^{n} x_i^2 \left(1 + \frac{(SS^2 - 1) x_j{}^2 + 2SS^2 x_j + SS^2}{\sum_{i=1}^{n} x_i^2}\right)} \times \sqrt{\sum_{i=1}^{n} y_i^2 \left(1 + \frac{(SS^2 - 1) y_j{}^2 + 2SS^2 y_j + SS^2}{\sum_{i=1}^{n} y_i^2}\right)}}$$

$$= \frac{(SS^2 - 1) y_j z_j + SS^2 y_j + SS^2 z_j + SS^2}{\|Z\| \times \|Y\| \times \sqrt{1 + \frac{(SS^2 - 1) z_j{}^2 + 2SS^2 z_j + SS^2}{\sum_{i=1}^{n} z_i^2}} \times \sqrt{1 + \frac{(SS^2 - 1) y_j{}^2 + 2SS^2 y_j + SS^2}{\sum_{i=1}^{n} y_i^2}}} + \cos(x)$$

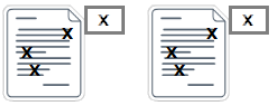
$$\times \frac{1}{\sqrt{1 + \frac{(SS^2 - 1) z_j{}^2 + 2SS^2 z_j + SS^2}{\sum_{i=1}^{n} z_i^2}} \times \sqrt{1 + \frac{(SS^2 - 1) y_j{}^2 + 2SS^2 y_j + SS^2}{\sum_{i=1}^{n} y_i^2}}}$$

Observa-se que $\cos(a)$ está dependente de:

- $\cos(x)$ - cosseno do ângulo entre os documentos antes da integração das *tags*;
- y_j *e* z_j - frequência correspondente em cada um dos documentos antes da integração das *tags*;
- $\|Y\|$ *e* $\|Z\|$ – Norma dos vetores de cada documento antes da integração;
- SS - parâmetro que permite controlar a importância dada à *tag* associada ao documento.

Portanto, apresenta-se de seguida a análise das situações em que ambos os documentos têm *tags* associadas e em que as *tags* aparecem em ambos os documentos com a mesma frequência.

- **A *tag* aparece no documento mais do que uma vez**



Observa-se pela análise da Figura 43 que, à medida que SS aumenta, ou dito de outro modo, quando é dada mais importância às *tags*, verifica-se que cosseno do ângulo tende a aproximar-se de 1, independentemente dos documentos estarem próximos ou afastados antes da integração. Isto significa que o ângulo formado entre os documentos tende a aproximar-se de zero.

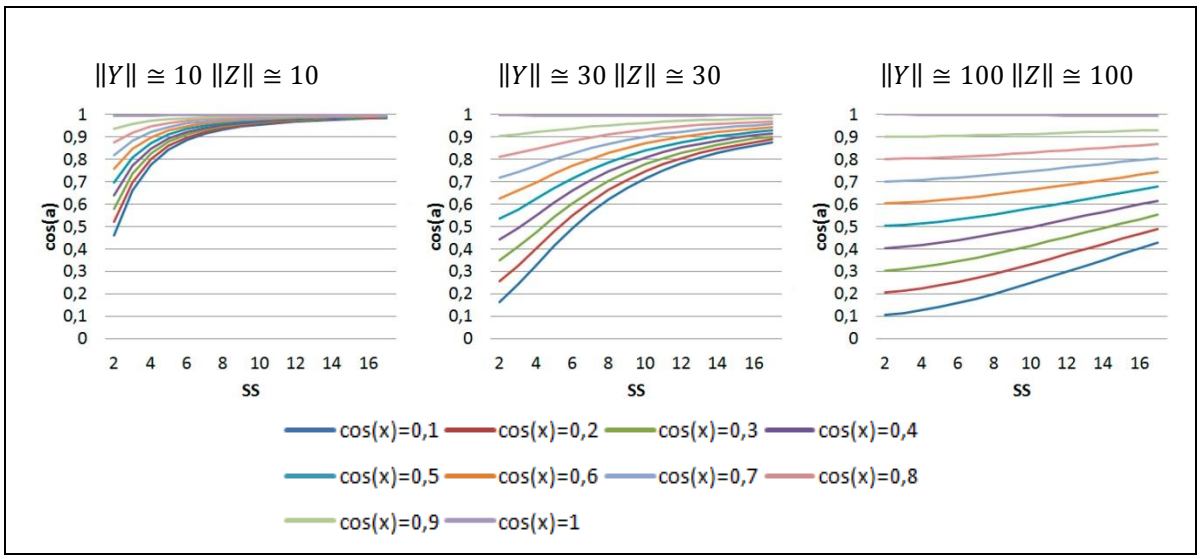


Figura 43: Variação do cos(a) quando a *tag* aparece nos dois documentos mais do que uma vez.

- **A *tag* não aparece no documento**

Neste caso, observa-se que a influência das *tags* tende a ser menos significativa do que quando a *tag* aparece no documento mais do que uma vez (Figura 44).

Note-se que só se verifica uma rápida tendência para aproximar de forma significativa os documentos quando temos normas dos vetores iniciais muito baixas. Nos restantes casos, tornam-se necessários valores de SS significativamente mais elevados. Por

exemplo, se os documentos tiverem aproximadamente norma 100, apenas quando SS toma valores próximos de 11 se torna visível uma fraca aproximação, ainda que apenas para os documentos que inicialmente estavam mais afastados.

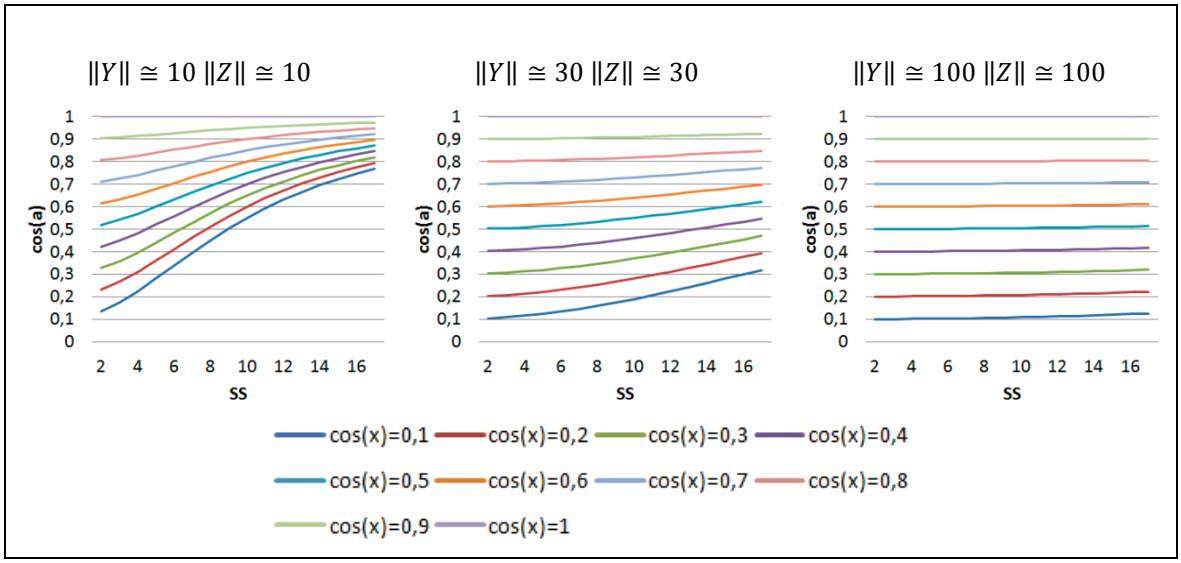
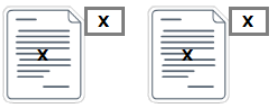


Figura 44: Variação do cos(a) quando a *tag* não aparece nos dois documentos.

- **A *tag* aparece uma vez nos documentos**



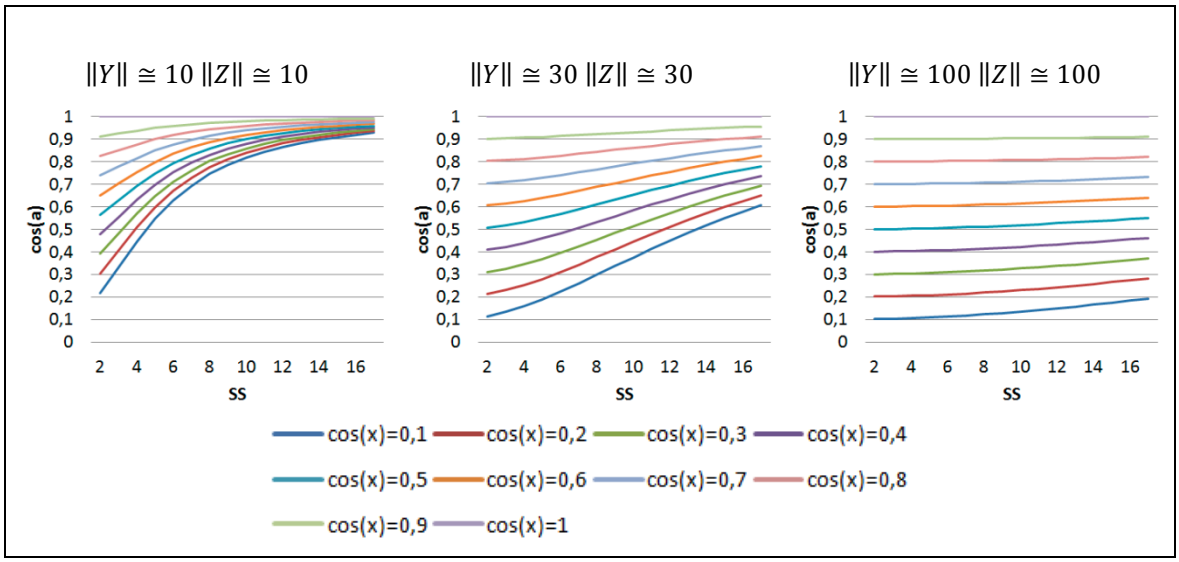


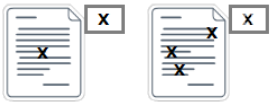
Figura 45: Variação do cos(a) quando a *tag* aparece nos dois documentos uma única vez.

Como nesta situação a tag aparece uma vez no texto, verifica-se que em relação à situação analisada anteriormente existe uma aproximação mais significativa entre os documentos, pelo menos para os que apresentam antes da integração das tags uma

norma inferior a 30 (Figura 45). Note-se ainda que apesar de não produzir resultados significativos para quando a norma é 100, verifica-se uma aproximação entre os documentos para valores de SS superiores a 6, no que respeita a documentos que inicialmente apresentam cosseno do ângulo muito baixos.

3.2.2. Documentos que partilham as mesmas *tags* mas as *tags* ocorrem nos dois documentos com frequências diferentes.

- **A *tag* aparece num dos documentos uma vez e no outro mais do que uma vez.**



Se $z_j = 1$ e $y_j \geq 2$ então o cosseno do ângulo dos novos vetores é:

$$\cos(a) = \frac{(2SS^2 - 1) \times y_j + 2SS^2 + 2SSy_j + 2SS}{\|Z\| \times \|Y\| \times \sqrt{\frac{[(SS+1)^2 - 1]{y_j}^2 + 2(SS+1)^2 y_j + (SS+1)^2}{\sum_{i=1}^{n} {y_i}^2}} + 1 \times \sqrt{\frac{(SS^2 - 1) + 2SS^2 + SS^2}{\sum_{i=1}^{n} {z_i}^2}} + 1} \quad \text{Equação 17}$$

$$+$$

$$\cos(x) \times \frac{1}{\sqrt{\frac{[(SS+1)^2 - 1]{y_j}^2 + 2(SS+1)^2 y_j + (SS+1)^2}{\sum_{i=1}^{n} {y_i}^2}} + 1 \times \sqrt{\frac{(SS^2 - 1) + 2SS^2 + SS^2}{\sum_{i=1}^{n} {z_i}^2}} + 1}$$

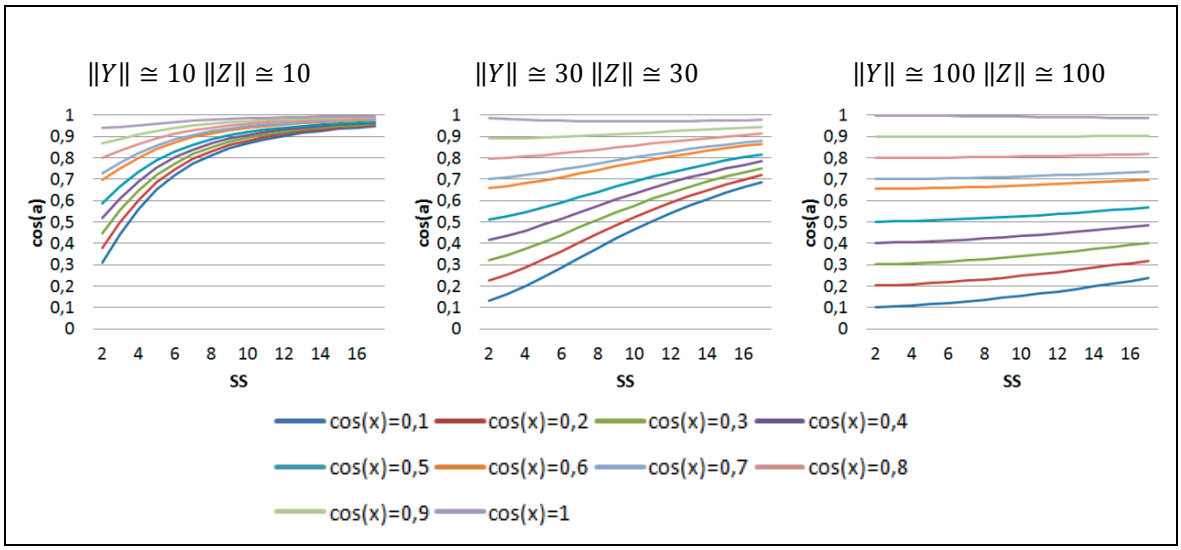
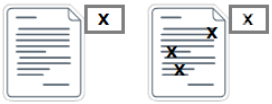


Figura 46: Variação do cos(a) quando a *tag* aparece num documento uma vez e no outro mais do que uma vez.

Nesta situação verifica-se que para os primeiros valores de SS, os documentos que

incialmente estavam muito próximos passam a estar afastados e os documentos que estavam muito afastados tendem a aproximar-se mais rapidamente assim que SS aumenta (Figura 46).

- **A *tag* não aparece num dos documentos e no outro aparece mais do que uma vez.**



Se $z_j = 0$ e $y_1 \geq 2$ então o cosseno do ângulo dos novos vetores é:

$$\cos(a) = \qquad \text{Equação 18}$$

$$= \frac{SS \times (SS+2) \times (y_j + 1)}{\|Z\| \times \|Y\| \times \sqrt{\frac{[(SS+2)^2 - 1]{y_j}^2 + 2(SS+2)^2 y_j + (SS+2)^2}{\sum_{i=1}^{n} {y_i}^2} + 1} \times \sqrt{\frac{SS^2}{\sum_{i=1}^{n} {z_i}^2} + 1}} +$$

$$\cos(x) \times \frac{1}{\sqrt{\frac{[(SS+2)^2 - 1]{y_j}^2 + 2(SS+2)^2 y_j + (SS+2)^2}{\sum_{i=1}^{n} {y_i}^2} + 1} \times \sqrt{\frac{SS^2}{\sum_{i=1}^{n} {z_i}^2} + 1}}$$

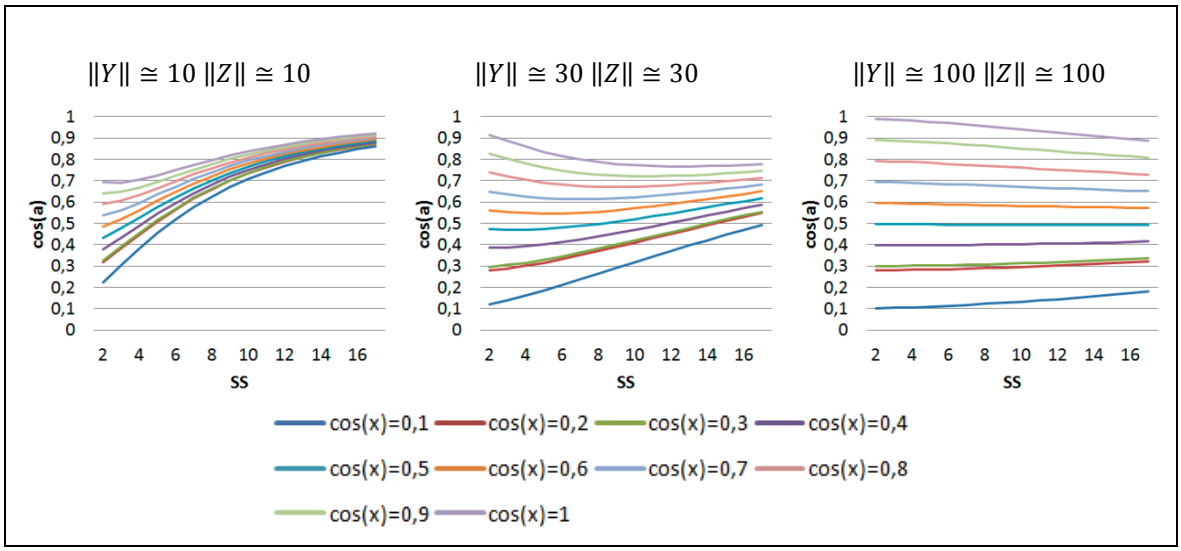


Figura 47: Variação do cos(a) quando a *tag* não aparece num dos documento e no outro aparece mais do que uma vez.

Em comparação com a situação anterior, constata-se que as curvas descrevem comportamento idêntico mas com um afastamento maior entre os documentos que estavam inicialmente muito próximos e uma aproximação mais reduzida entre os documentos que inicialmente se encontravam muito afastados. À medida que a norma

aumenta também se verifica que são necessários valores de SS cada vez mais elevados para permitir que os documentos se aproximem de cos(a) igual a 1.

- **A *tag* aparece num dos documentos uma vez e no outro não aparece**

Se $z_j = 0$ e $y_j = 1$ então o cosseno do ângulo dos novos vetores é:

$$\cos(a) = \qquad \text{Equação 19}$$

$$= \frac{SS \times (SS+1) \times (y_j + 1)}{\|Z\| \times \|Y\| \times \sqrt{\frac{[(SS+1)^2 - 1]{y_j}^2 + 2(SS+1)^2 y_j + (SS+1)^2}{\sum_{i=1}^{n} {y_i}^2}} + 1 \times \sqrt{\frac{SS^2}{\sum_{i=1}^{n} {z_i}^2} + 1}} +$$

$$\cos(x) \times \frac{1}{\sqrt{\frac{[(SS+1)^2 - 1]{y_j}^2 + 2(SS+1)^2 y_j + (SS+1)^2}{\sum_{i=1}^{n} {y_i}^2}} + 1 \times \sqrt{\frac{SS^2}{\sum_{i=1}^{n} {z_i}^2} + 1}}$$

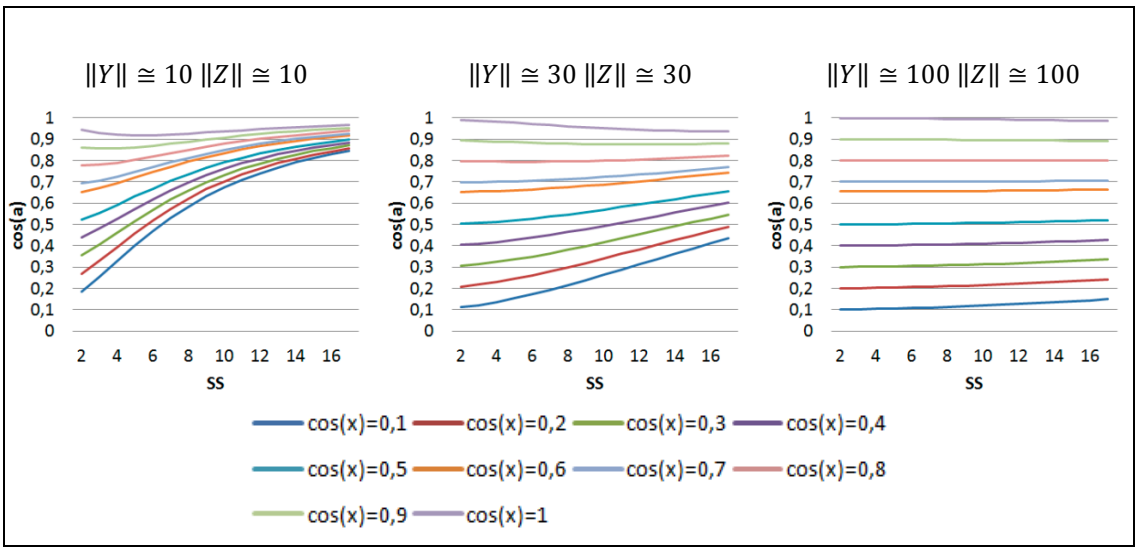


Figura 48: Variação do cos(a) quando a *tag* não aparece num dos documento e no outro aparece uma única vez.

Observando a Figura 48 verifica-se para os primeiros valores de SS um afastamento dos documentos que inicialmente estavam muito próximos, ainda que de forma mais ténue do que nas duas situações analisadas anteriormente. A partir de determinados valores de SS verifica-se que os documentos tendem a ficar muito próximos. Para norma 10, a partir de SS=5, todas as curvas voltam a ser crescentes, sendo contudo visível que para norma

100, quase não se detetam alterações para os valores de SS analisados, tanto nesta situação como nas duas anteriores.

3.2.3. Documentos que não partilham a mesma *tag*

O cosseno do ângulo entre dois documentos que não partilham as mesma *tags* é dado por:

$$\cos(a) = \qquad \text{Equação 20}$$

$$= \frac{(SS-1)y_1 \times z_1 + SSz_1}{\|Z\| \times \|Y\| \times \sqrt{\frac{(SS^2-1){y_1}^2 + 2SS^2 y_1 + SS^2}{\sum_{i=1}^{n} {y_i}^2} + 1}} +$$

$$\cos(x) \times \frac{1}{\sqrt{\frac{(SS^2-1){y_1}^2 + 2SS^2 y_1 + SS^2}{\sum_{i=1}^{n} {y_i}^2} + 1}}$$

De seguida apresentamos a relação de documentos cuja *tag* aparece mais do que uma vez no texto com documentos que não têm essa *tag* associada:

- **A *tag* aparece uma vez no documento que não tem essa *tag* associada**

Na Figura 49 apresentam-se 3 gráficos, onde se exploram as diferenças na variação do cosseno do ângulo entre dois documentos, cuja norma dos vetores dos documentos pode variar entre 10 e 100.

Nota: para a elaboração dos gráficos considerou-se que a *tag* aparece 3 vezes no documento a que foi associada.

Observe-se que à medida que SS aumenta, o cosseno do ângulo dos documentos decresce, ou seja, o ângulo entre os documentos fica maior. Note-se contudo que, à medida que a norma dos vetores aumenta, o cosseno do ângulo só começa a ser modificado para valores de SS cada vez maiores.

Quanto maior for o número de vezes que a *tag* aparece no documento a que foi associada, mais rapidamente ficam os documentos afastados (à medida que SS aumenta).

Para potenciar o afastamento dos documentos, pode sugerir-se que a coordenada do vetor a quem não foi associada a *tag* seja considerada zero, que no caso permite apenas um decréscimo ligeiro do cosseno do ângulo.

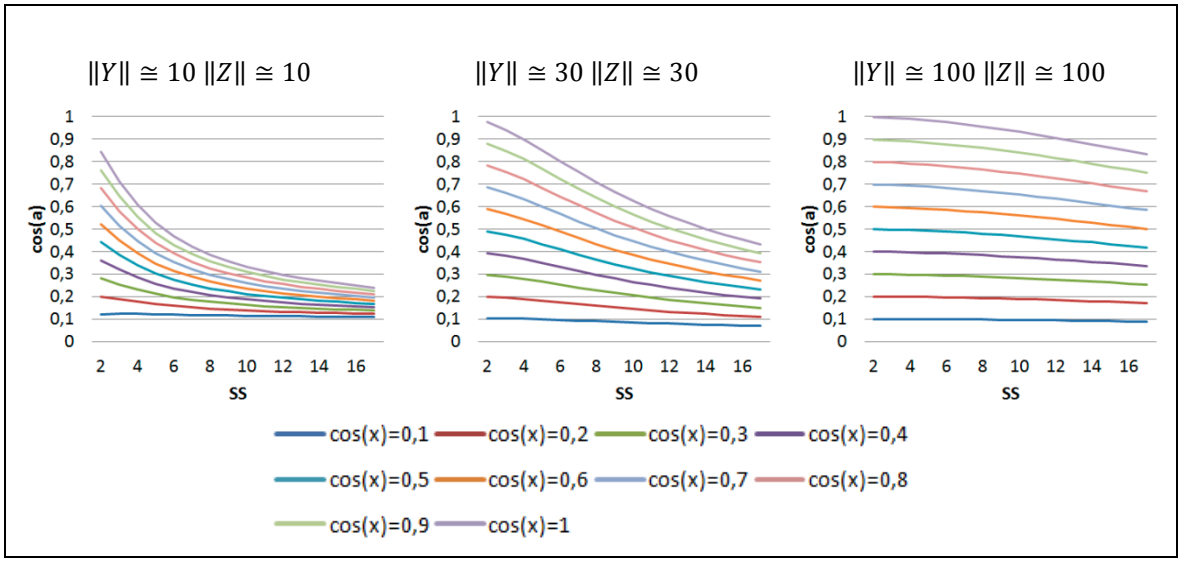
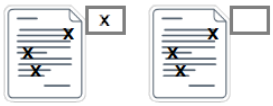


Figura 49: Variação do cos(a) quando a *tag* aparece uma vez no documento que não tem essa *tag* associada e no outro aparece mais do que uma vez.

- **A *tag* aparece mais do que uma vez no documento que não tem essa *tag* associada**



Para analisar como varia o cosseno do ângulo entre dois documentos que não têm a mesma *tag* associada mas que aparece nos dois documentos mais do que uma vez, considerou-se que a *tag* aparece 3 vezes no documento a que foi associada e 5 vezes no outro documento.

Como se pode verificar pelos resultados apresentados na Figura 50, quando a norma dos documentos está próxima de 10 constata-se que se o cosseno do ângulo inicial for baixo os vetores dos documentos tendem a aproximar-se à medida que SS aumenta (ainda que não seja suficiente para influenciar o agrupamento dos dois documentos no mesmo *cluster*). Analisando os restantes gráficos verifica-se que quando a norma aumenta, a tendência é a de que o cosseno do ângulo decresça (quando SS aumenta).

Note-se ainda que caso se pretenda reduzir a frequência com que aparece a *tag* no documento a que não foi associada podemos verificar, tal como mostra a Figura 51, que os documentos se afastam à medida que SS aumenta mas não de forma significativa.

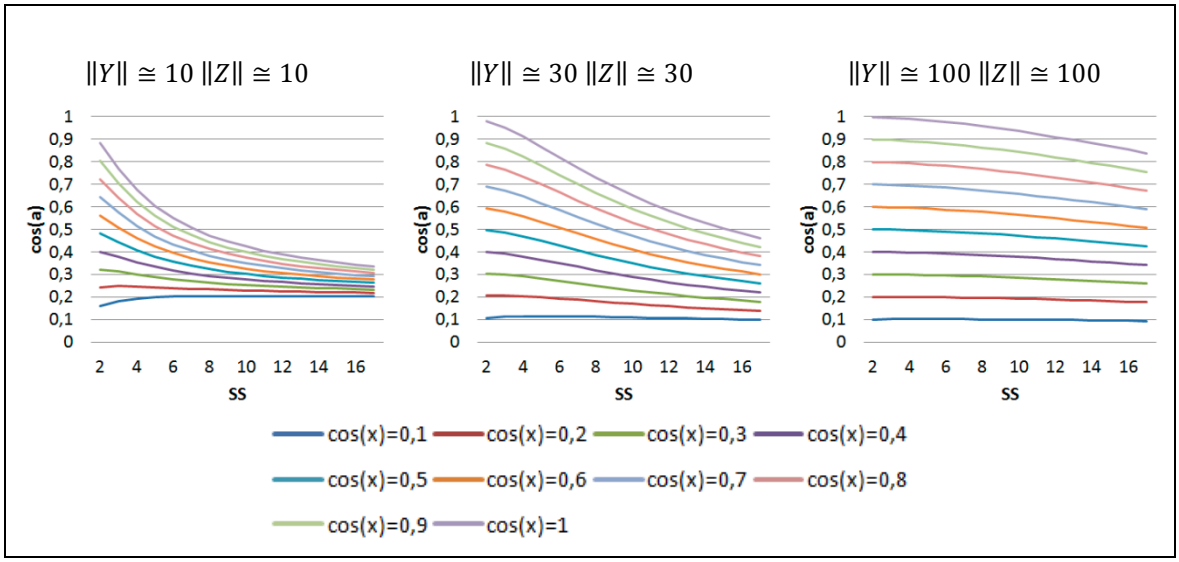


Figura 50: Variação do cos(a) quando a *tag* aparece mais do que uma vez no documento que não tem essa *tag* associada e no outro aparece mais do que uma vez.

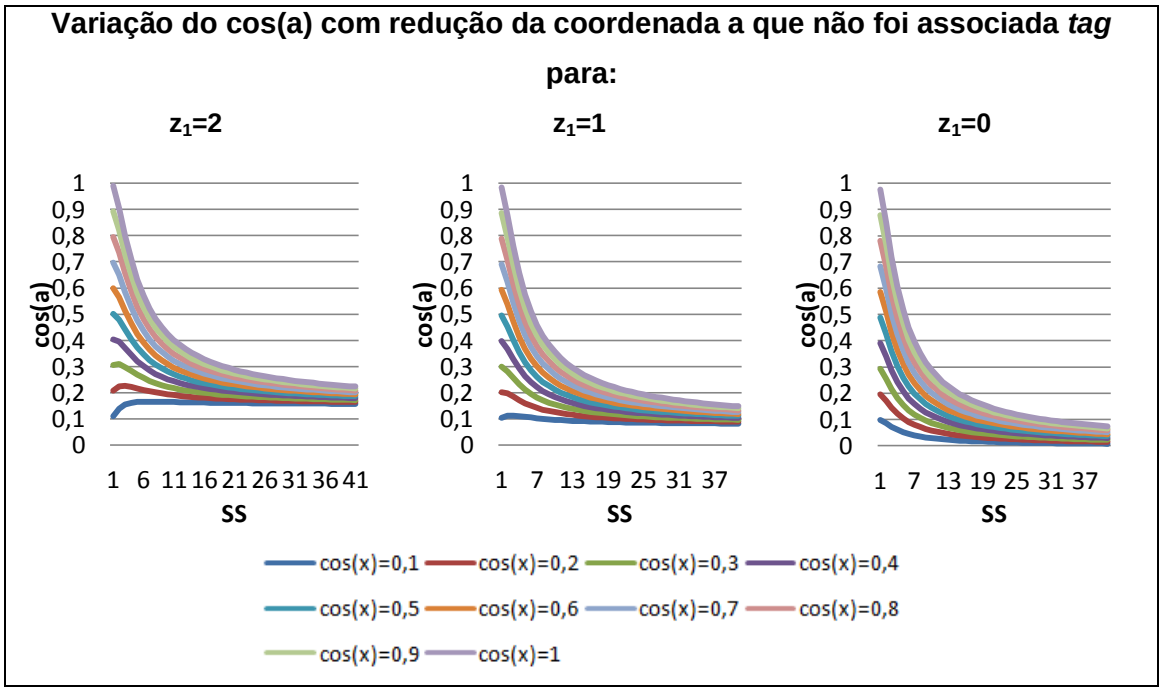


Figura 51: Variação do cos(a) para diferentes frequências da *tag* no documento ao qual não foi associada.

De referir ainda que para documentos com norma próxima de 100 são necessários valores de SS superiores a 17 para ser possível vermos alterações significativas entre os ângulos dos documentos, tal como mostra a Figura 52.

Por fim é necessário salientar que, tal como na situação analisada anteriormente, quanto maior for o número de vezes que a *tag* aparece no documento a que foi associada mais rapidamente são afastados os documentos à medida que k aumenta.

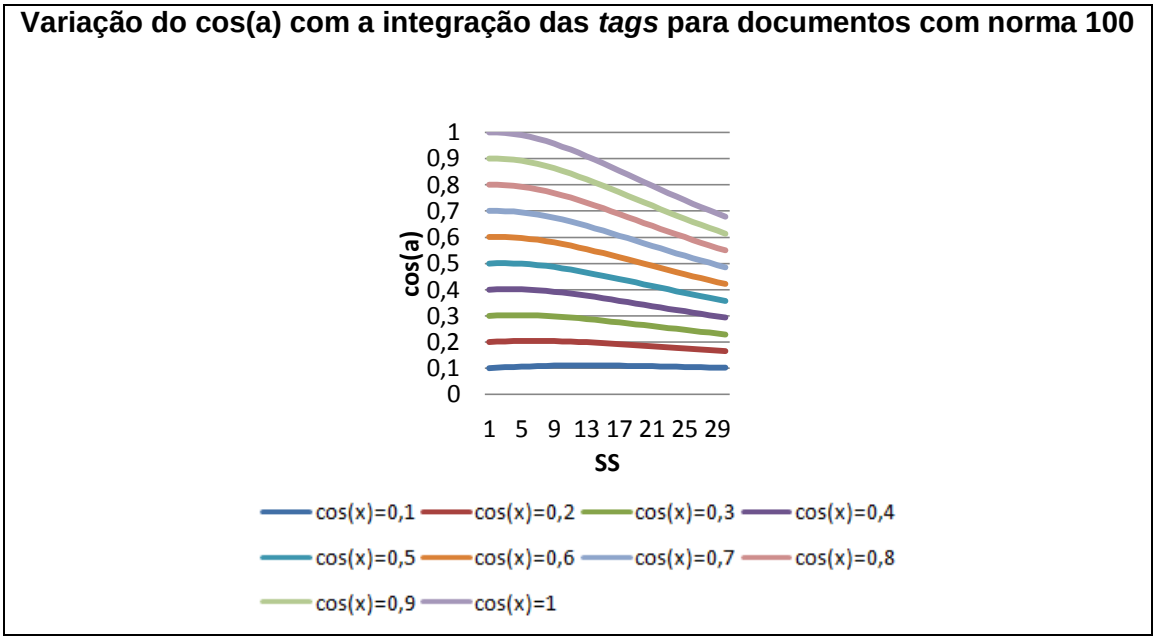


Figura 52: Determinação do parâmetro SS que permite alterar o ângulo entre os documentos quando a norma é próxima de 100 – Situação 1.

- **A *tag* não aparece no documento que não tem essa *tag* associada**

Para a elaboração da Figura 53, considerou-se que a *tag* aparece 3 vezes no documento a que foram associadas as *tags*.

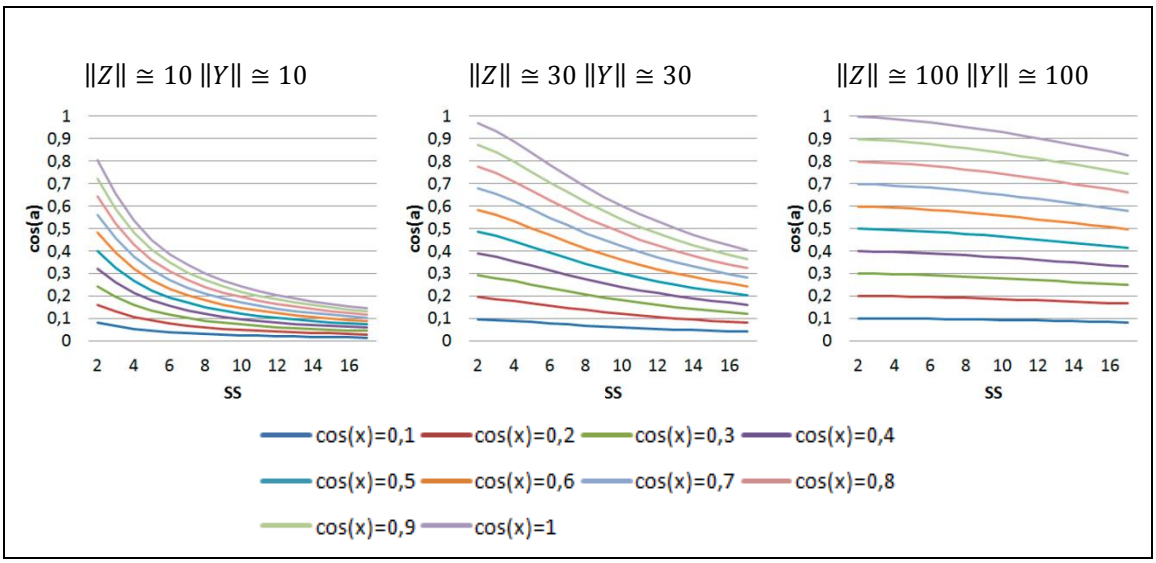


Figura 53: Variação do cos(a) quando a *tag* não aparece no documento que não tem essa *tag* associada e no outro aparece mais do que uma vez.

Estes gráficos (Figura 53) revelam que os documentos são significativamente afastados

quando aumentamos o valor de SS. Sendo que, para documentos com norma próxima de 100 podem ser tomados valores de SS superiores de 17, se pretendermos ver afastados mais significativamente os documentos que estavam muito próximos inicialmente.

Resta referir que, tal como nas situações analisadas anteriormente, quando maior for a frequência da *tag* no documento a que foi associada, mais rapidamente ficam os documentos afastados quando se aumenta o SS.

3.2.4. Relação de documentos cuja *tag* aparece uma vez no texto com documentos que não têm essa *tag* associada

- **A *tag* não aparece no documento que não tem essa *tag* associada**

Para a situação apresentada na Figura 54, verifica-se que existe sempre um afastamento entre os documentos, sendo que para normas mais elevadas dos documentos em análise é necessário permitir que o SS possa variar para valores superiores a 17, com a finalidade de permitir que seja significativo o afastamento entre os documentos.

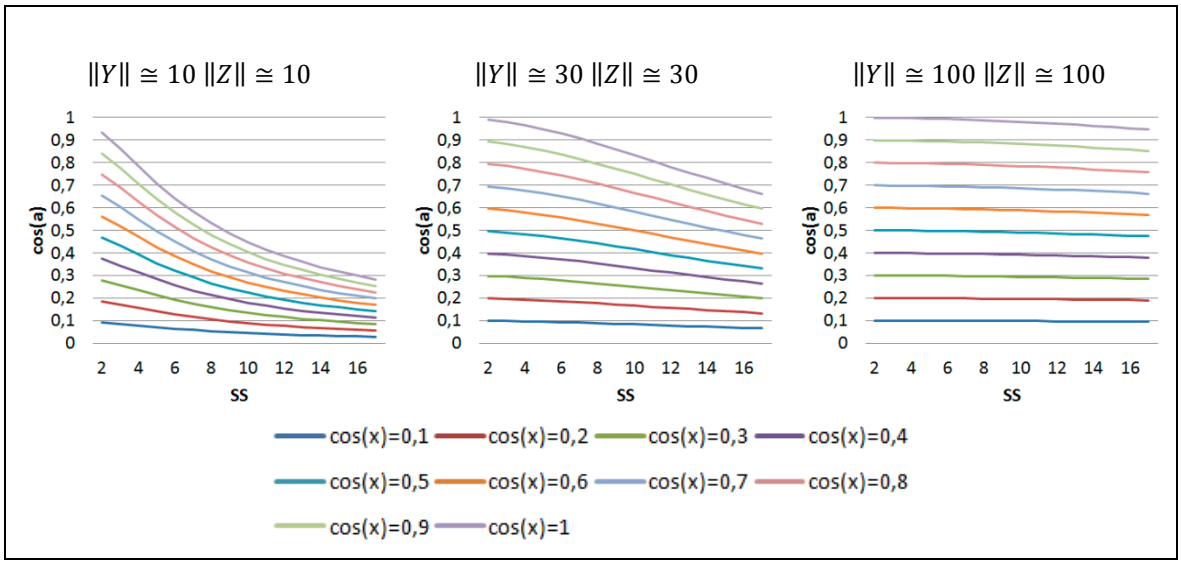
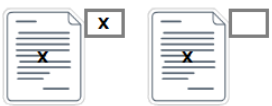


Figura 54: Variação do cos(a) quando a *tag* não aparece no documento que não tem essa *tag* associada e no outro aparece uma vez.

- **A *tag* aparece uma vez no documento que não tem essa *tag* associada**



Da análise da Figura 55 verifica-se que um afastamento entre os documentos que não partilham a mesma *tag* em que a mesma aparece nos dois documentos apenas uma vez. Tal como na situação analisada anteriormente quando a norma dos documentos em causa aumenta para valores próximos de 100, verifica-se que as alterações só são produzidas para valores de SS superiores a 10, e que inicialmente estavam muito próximos.

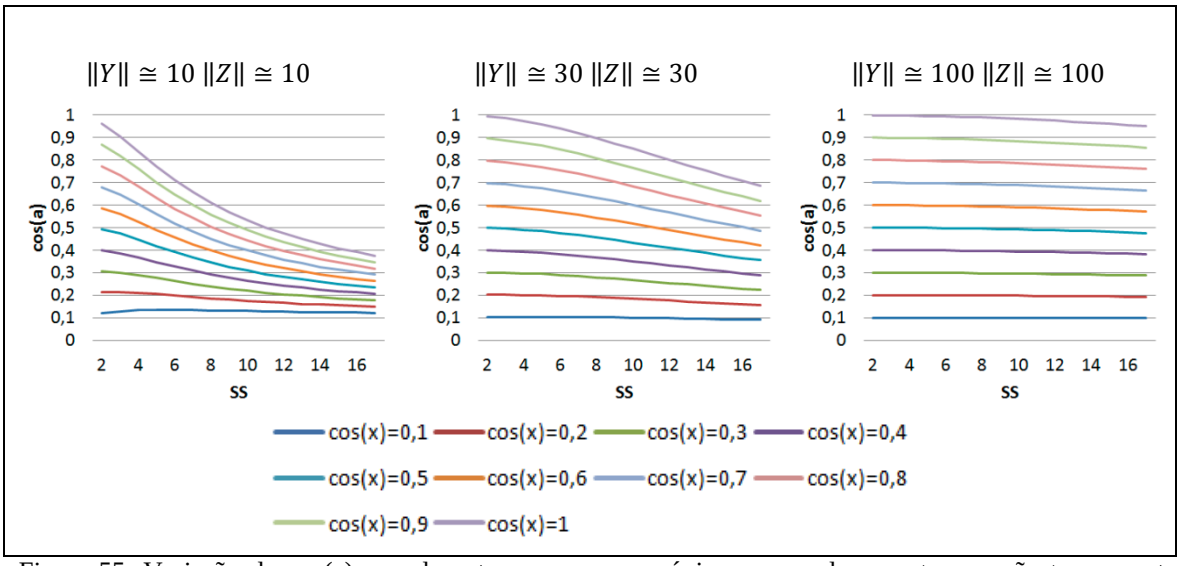
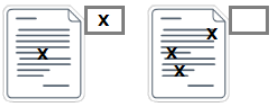


Figura 55: Variação do cos(a) quando a *tag* aparece uma única vez no documento que não tem essa *tag* associada e no outro aparece uma vez.

Para a situação apresentada e no sentido de potenciar o afastamento entre os documentos, apenas se pode considerar que o documento que não tem tag associada passa a não considerar a *tag* no documento. De qualquer modo, o afastamento resultante desta manipulação não é significativo.

- **A *tag* aparece mais do que uma vez no documento que não tem essa *tag* associada**



Para esta situação considerou-se que no documento a que não foi associada a *tag*, esta aparece 3 vezes no documento.

Para a situação particular apresentada (Figura 56) observa-se que para normas baixas (norma 10), documentos que inicialmente estavam muito afastados tendem a aproximar-se ligeiramente quando se aumenta o valor de SS e documentos que estavam muito próximos antes da integração das *tags* tendem a afastar-se rapidamente para os mesmos valores de SS.

Observe-se ainda que, quanto maior for a norma dos documentos em análise, menos significativa será a aproximação e o afastamento dos documentos nas mesmas circunstâncias das descritas anteriormente, sendo necessário, por exemplo para norma 100, que SS possa variar para valores superiores a 17.

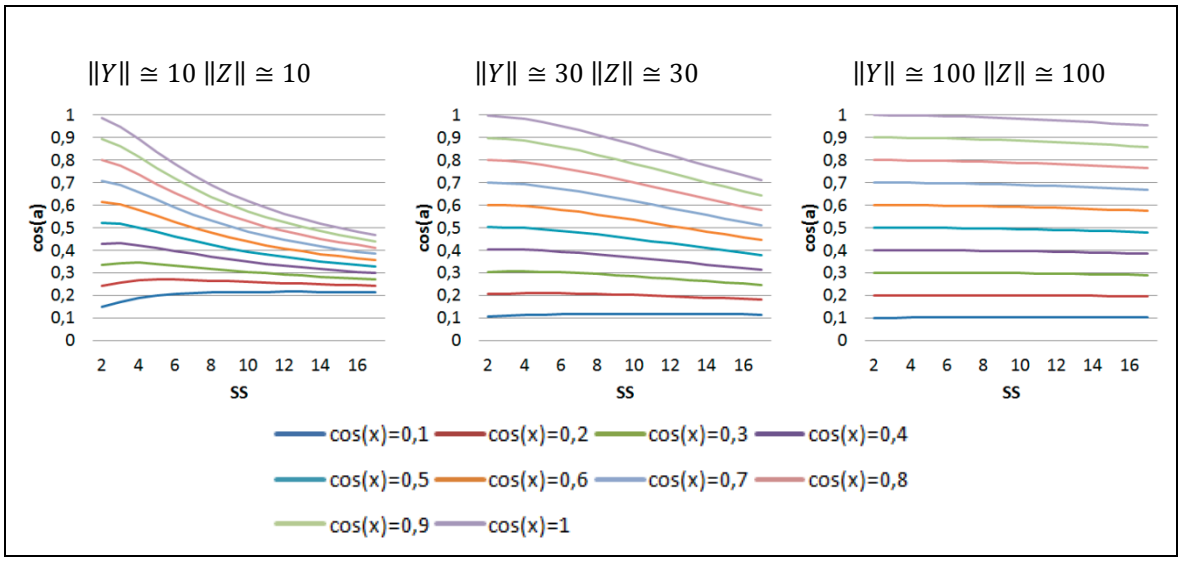


Figura 56: Variação do cos(a) quando a *tag* aparece mais do que uma vez no documento que não tem essa *tag* associada e no outro aparece uma vez.

Tal como nas situações analisadas, quando se reduz para zero, um ou dois a coordenada referente à *tag* que aparece no documento a que não foi associada, constata-se uma aproximação dos documentos, que é tanto mais significativa quanto maior for a frequência da *tag* nesse documento. Para a situação específica apresentada, como a *tag* aparece 3 vezes, quando se reduz para dois, um ou zero, observam-se diferenças mas pouco significativas.

3.2.5. Relação de documentos cuja *tag* não aparece no texto com documentos que não têm essa *tag* associada

- **A *tag* não aparece no documento que não tem essa *tag* associada**

Na presente situação (Figura 57) pode constatar-se que para documentos com normas próximas de 10 os documentos afastam-se significativamente quando se faz variar SS até 17.

No entanto, o mesmo não acontece quando a norma passa a ser 30 ou 100. No primeiro caso, observa-se um afastamento dos documentos que estavam mais próximos antes da

integração das *tags* mas para valores de SS mais elevados e de forma menos significativa do que para normas próximas de 10.

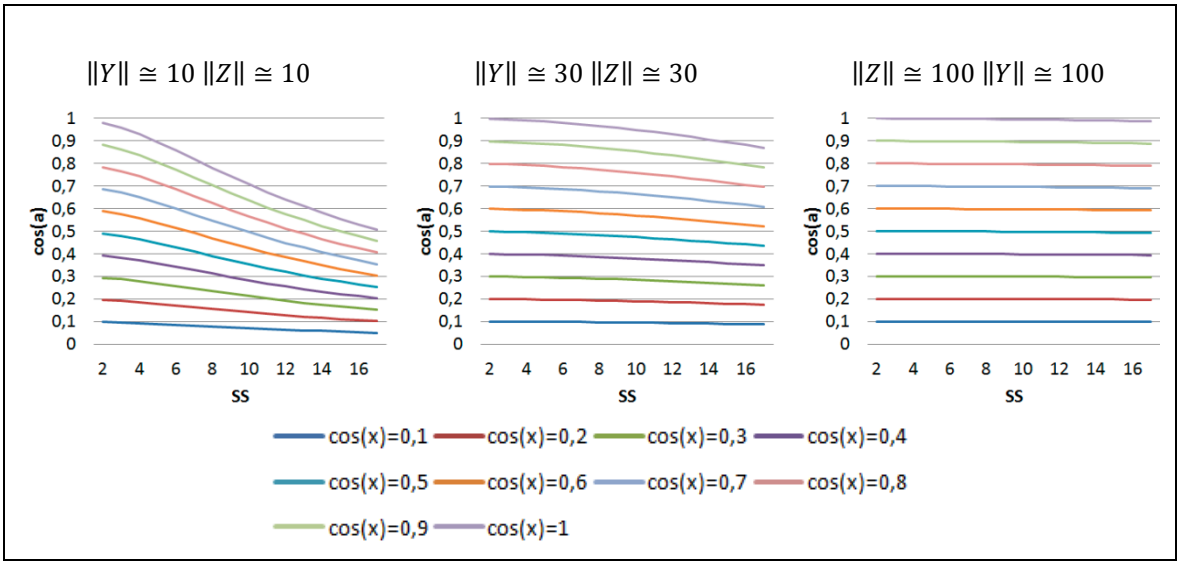


Figura 57: Variação do cos(a) quando a *tag* não aparece nem no documento que não tem essa *tag* associada nem no outro documento.

Quando a norma dos documentos está próxima de 100 não se registam alterações quando se faz variar SS até 17 (Figura 58). No entanto, como se pode ver pelo gráfico apresentado abaixo quando aumentamos o SS, é possível observar alterações no que respeita ao afastamento dos documentos, sobretudo os que permaneciam mais próximos inicialmente.

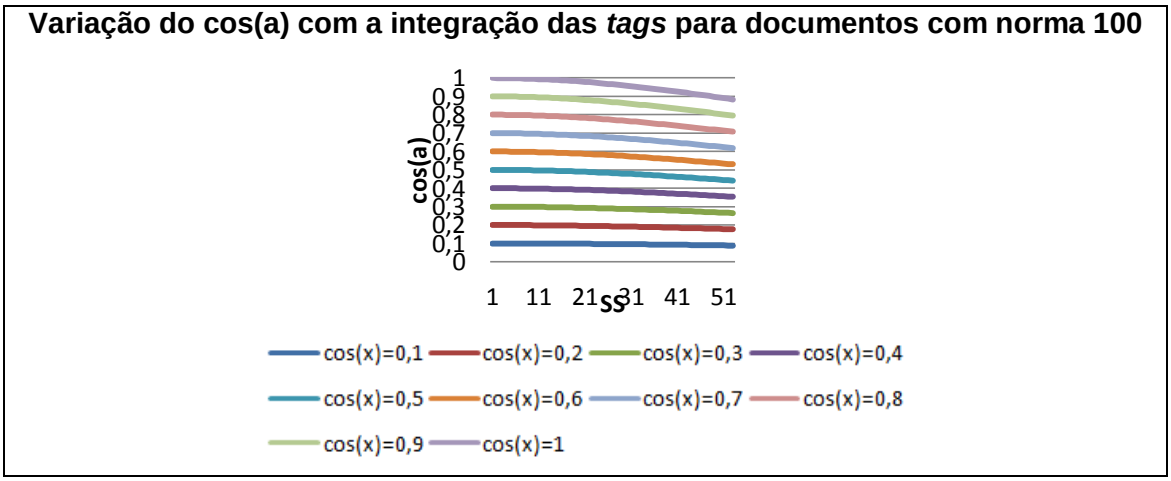
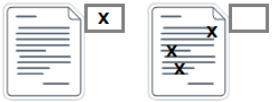


Figura 58: Determinação do parâmetro SS que permite alterar o ângulo entre os documentos quando a norma é próxima de 100 – Situação 2.

- **A *tag* aparece mais do que uma vez no documento que não tem essa *tag* associada**



Para a construção dos presentes gráficos, considerou-se que a *tag* aparece três vezes no documento a que não foi associada.

Tal como nas situações analisadas anteriormente, à medida que a norma dos documentos aumenta, menos significativo é o afastamento entre os documentos quando SS aumenta (Figura 59). Do mesmo modo, é necessário fazer variar SS para valores superiores de modo a ser visível o afastamento entre os documentos, como se pode constatar no gráfico apresentado na Figura 60.

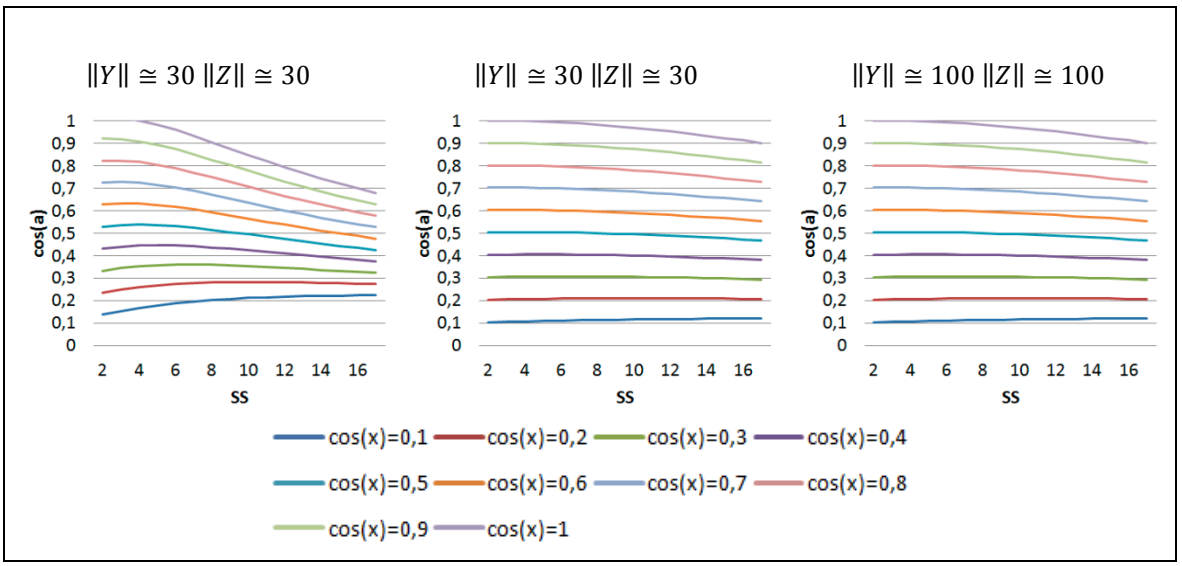


Figura 59: Variação do cos(a) quando a *tag* aparece mais do que uma vez no documento que não tem essa *tag* associada e no outro nunca aparece.

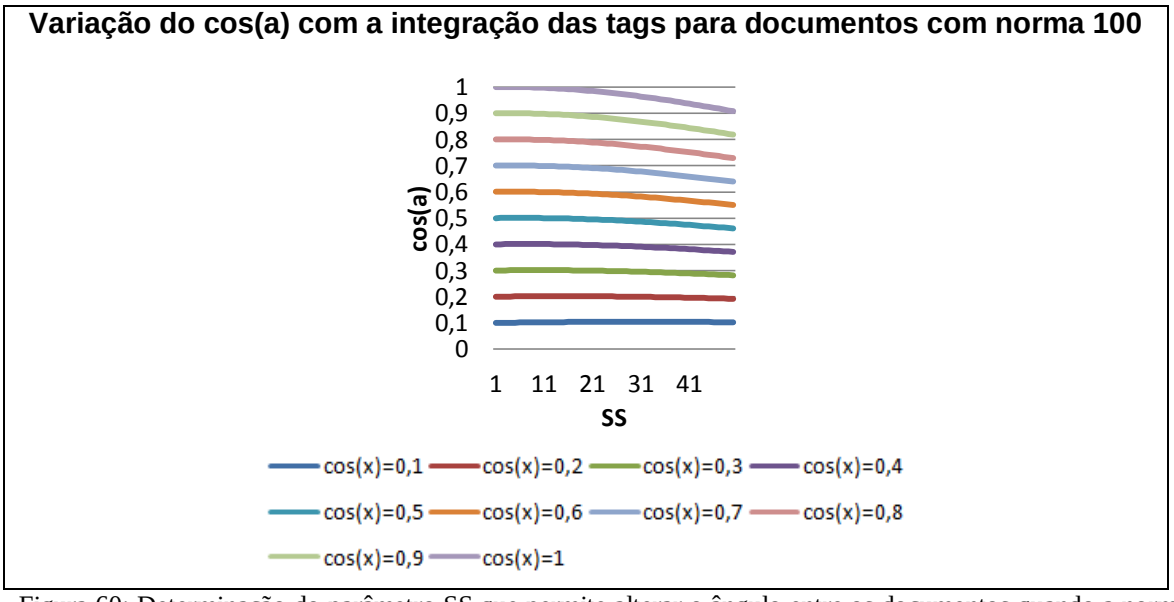
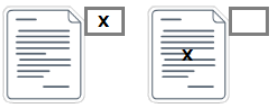


Figura 60: Determinação do parâmetro SS que permite alterar o ângulo entre os documentos quando a norma é próxima de 100 – Situação 3.

- **A *tag* aparece uma vez no documento que não tem essa *tag* associada**



A última situação em análise (Figura 61), evidencia tal como nas situações anteriores um maior afastamento entre os documentos quando a sua norma está próxima de 10.

Além disso, é igualmente necessário aumentar o valor de SS para ser possível detetar alterações no que respeita ao afastamento entre os documentos.

Na situação em análise e com vista a potenciar o afastamento mais rápido entre os documentos à medida que SS aumenta, apenas é possível reduzir a frequência com que a *tag* aparece no documento a que não foi associada de um para zero, sendo contudo de referir que esse afastamento não é significativamente relevante.

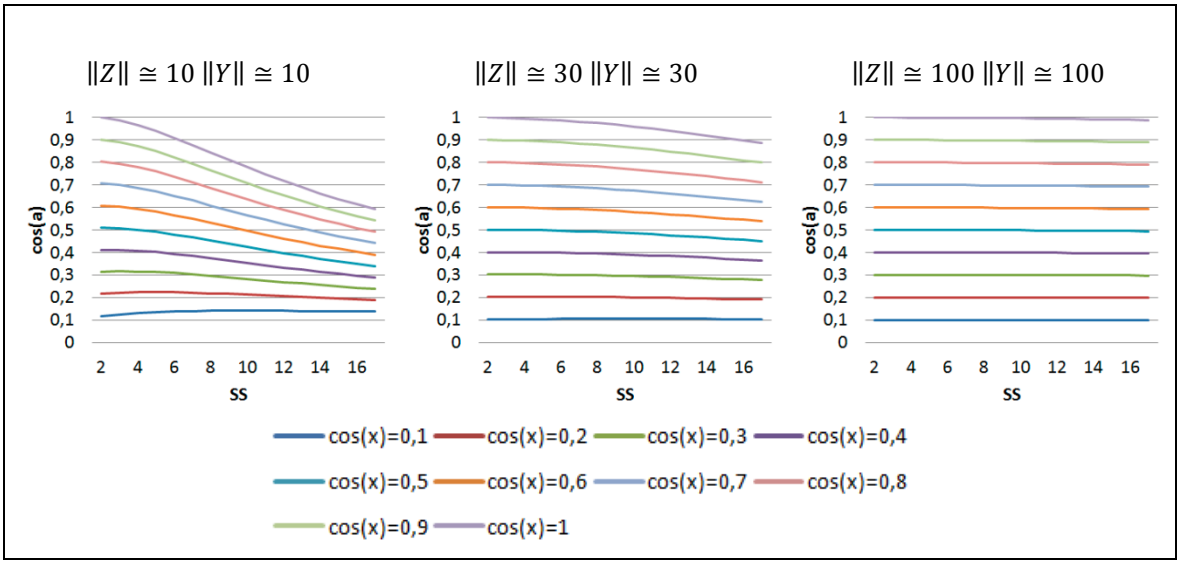


Figura 61: Variação do cos(a) quando a *tag* aparece uma vez no documento que não tem essa *tag* associada e no outro nunca aparece.

3.2.6. Síntese

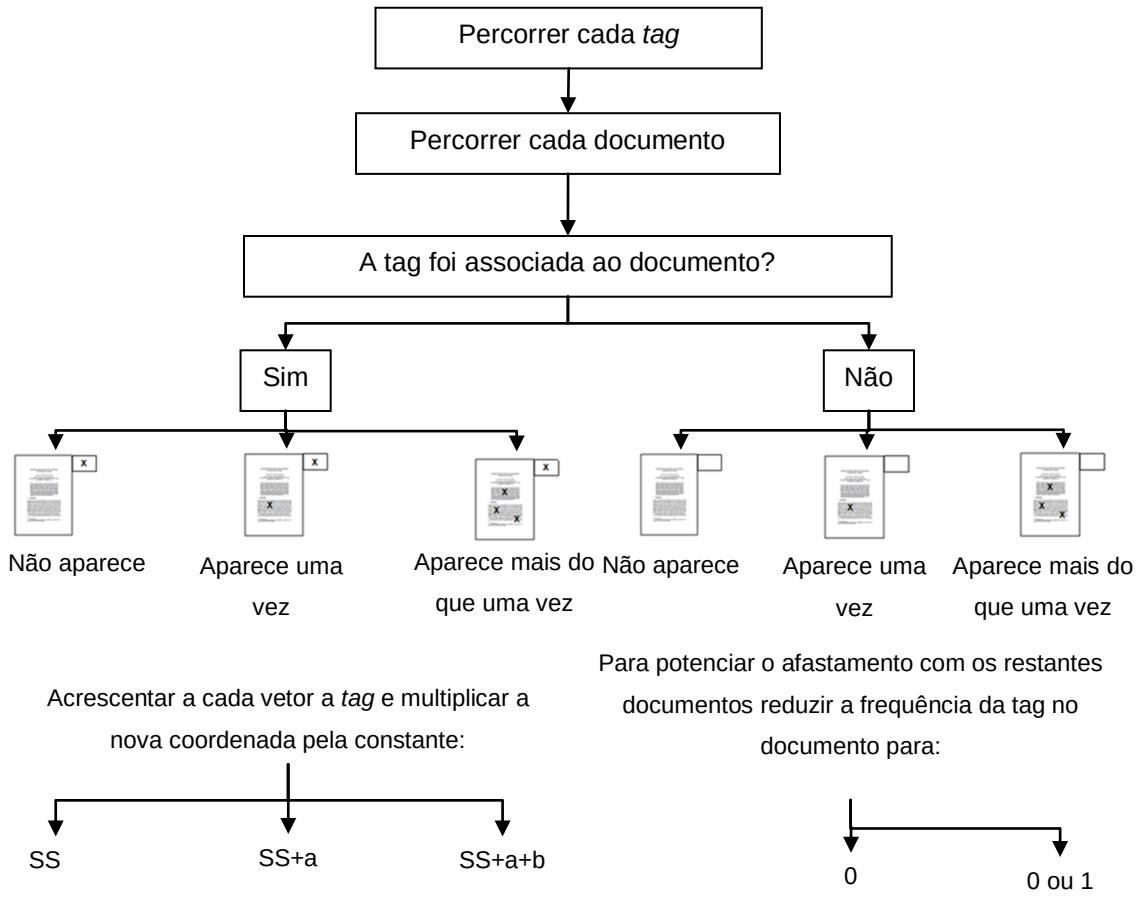
Para todas as situações apresentadas foram tomados em consideração documentos cuja norma se aproxima de 100 e valores de SS até 17, tendo-se verificado que a integração das *tags* não produz qualquer efeito no afastamento ou aproximação entre os documentos. São portanto necessários valores de SS até 60 a fim de ser possível observar algum afastamento ou aproximação.

Neste sentido, é importante salientar que foram feitos testes de contagens de palavras em alguns documentos tendo sido constatado que um documento que tenha como norma

100 poderá corresponder, em média, a documentos com 500 palavras diferentes. Além disso, foi feita também uma análise ao número de palavras diferentes que aparecem em notícias e a norma de cada vetor tende a ser inferior a 30.

Tendo em conta a análise efetuada e tomando como norma máxima dos documentos o valor 100, propõe-se que os valores de SS sejam decididos pelo utilizador sendo possível fazer variar o SS entre 0 (sem integração das *tags*) e SS=60. Além disso, o utilizador também pode decidir se quer potenciar o afastamento entre os documentos a que foram associadas *tags* e os restantes documentos, reduzindo a frequência com que a *tag* aparece nesses documentos.

No Esquema 1 apresenta-se o modelo de integração reajustado.



Esquema 1: Esquema reajustado do modelo de integração das *tags* no VSM

3.3. Algoritmo k-Communites (k-C)

A segunda abordagem para integrar as *tags* parte do princípio que as *tags* associadas aos documentos podem oferecer informações importantes sobre a forma como estes

estão relacionados. Aliado a este facto, e tendo por base a análise feita anteriormente, onde se sugere que a utilização da similaridade dos cossenos pode oferecer vantagens quando se pretende aumentar o peso dado às palavras que são coincidentes com as *tags* associadas, procura-se perceber o real impacto da utilização da similaridade dos cossenos e da distância Euclidiana no *clustering* de documentos usando o algoritmo *k-means*. Desta análise combinada, rede de *tags* versus medidas de similaridade, propomos um novo algoritmo de *clustering*: algoritmo *k-Communities* (k-C)

3.3.1. Reflexão sobre a Implementação do Algoritmo *k-means* com a Distância Euclidiana versus Similaridade dos Cossenos

Como já anteriormente referido, o modelo escolhido para implementar o algoritmo *k-means* é o *Vector Space Model* (VSM) considerando-se que cada documento é um vetor num determinado espaço de palavras (Salton, et al., 1975).

No VSM uma grande variedade funções de distância e similaridade têm sido usadas para calcular a similaridade entre os documentos, tais como a distância Euclidiana e a similaridade dos cossenos. Aliás, a medida de similaridade utilizada por defeito para implementar o algoritmo *k-means* é a distância Euclidiana. Contudo, a similaridade dos cossenos é também habitualmente utilizada.

De facto, a similaridade dos cossenos tem uma propriedade muito importante: a sua independência em relação ao comprimento do documento. Se tivermos dois documentos com exatamente os mesmos termos mas com frequências proporcionais, os documentos são tratados como iguais. Por exemplo: considerando $d_i = (2,0,1,0,3,0,0,1)$ e $d_j = (10,0,5,0,15,0,0,5)$ então $d_j = 5d_i$ e a similaridade dos cossenos destes dois documentos é 1, significando que o ângulo entre os dois documentos é 0 e consequentemente são vistos como idênticos. Uma aplicação prática desta situação poderá corresponder a dois documentos em que um corresponde a um resumo do outro, com vários termos comuns mas com frequências diferentes. Por outro lado, a distância Euclidiana entre d_i e d_j é aproximadamente 15,5, deixando claras as diferenças apontadas.

Como se pode ver, o algoritmo *k-means*, usando a distância Euclidiana (Figura 62) ou a similaridade dos cossenos (Figura 63), origina diferentes partições.

Observe-se, na Figura 62, que A e B são colineares com a origem do referencial, mas foram colocados em *clusters* diferentes. Contudo, na Figura 63, usando a similaridade dos cossenos, estes dois objetos são colocados no mesmo *cluster* porque o ângulo entre eles é zero e consequentemente estão à mesma distância dos dois centroides.

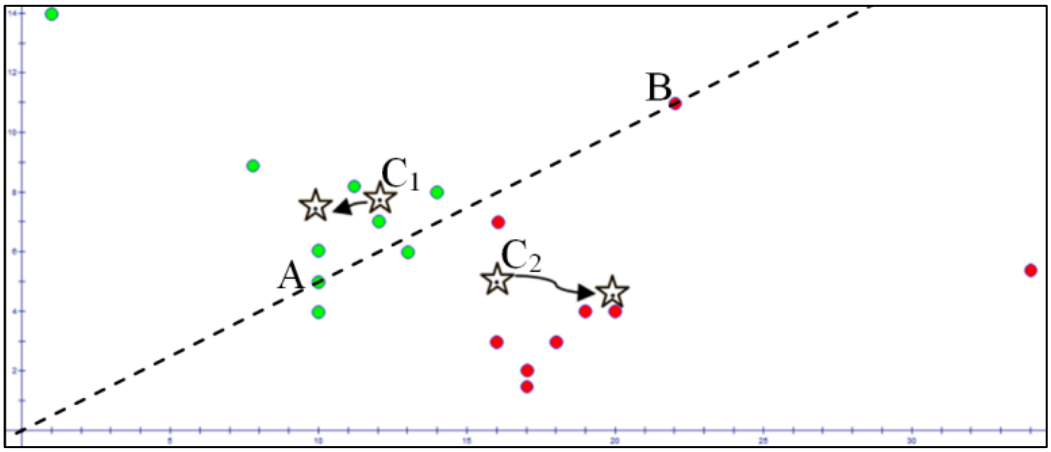


Figura 62: Algoritmo *k-means* usando a distância Euclidiana (Cunha, Figueira, & Mealha, 2013b).

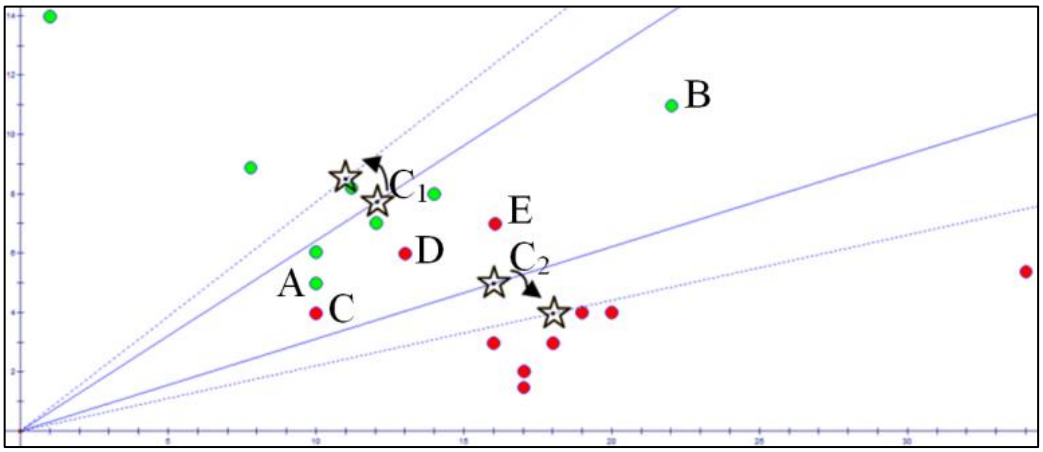


Figura 63: Algoritmo k-means usando a similaridade dos cossenos sem normalização dos vetores (Cunha, et al., 2013b).

Não obstante, a implementação do algoritmo *k-means* usando a similaridade dos cossenos não é garantia de eficácia. Por exemplo, C, D e E na Figura 63, estão mais próximos dos documentos colocados no *cluster* cuja semente é C1 do que dos documentos do seu *cluster*.

Quando escolhemos a similaridade dos cossenos para implementar o algoritmo *k-means* e caso a seleção dos novos centróides seja feita tendo em conta a distância Euclidiana, o novo centro estará, em média, à mesma distância (distância Euclidiana) de todos os documentos do *cluster*. Por consequência, os *outliers* (vistos sob a perspetiva da distância Euclidiana) irão influenciar a posição do novo centro e, como se pode ver na Figura 63, o ângulo entre o novo centro e C, D e E, respetivamente, é maior comparativamente com a semente inicial C1.

No sentido de colmatar este problema quando se implementa o algoritmo *k-means* usando a similaridade dos cossenos os vetores são normalizados, dando assim mais

importância à direção do que à magnitude (Zhong, 2005), assim o algoritmo *k-means* costuma denominar-se de *Spherical k-means*.

Contudo, pretendemos não normalizar os vetores e considerar como novo centro o vetor do documento mais semelhante aos restantes documentos do seu *cluster*, tendo em conta a similaridade dos cossenos.

3.3.2. Deteção de Comunidades para Selecionar as Sementes Iniciais

Criar algoritmos de *clustering* que sejam simultaneamente eficazes e eficientes continua a ser um desafio porque a qualidade dos *clusters* formados nem sempre é obtida através de algoritmos eficientes. O algoritmo *k-means* é reconhecido pela sua eficiência mas a qualidade dos *clusters* formados depende da seleção das sementes iniciais, porque a sua escolha aleatória pode resultar numa má otimização dos *clusters*. No sentido de melhorar a performance, vários métodos têm sido propostos. Entre eles está o algoritmo *k-means*++ proposto por Arthur e Vasilvitskii (2007), já visto no Capítulo 1, onde as sementes são selecionadas com probabilidades específicas antes de executar o algoritmo *k-means*, providenciando uma melhoria do número necessário de iterações até obter a convergência.

Contudo, o número de partições continua a ser um problema. Tendo em conta que os *clusters* formados devem satisfazer o interesse das pessoas, propomos a análise das *tags* associadas aos documentos. Assim, a deteção de comunidades será usada para ver como a informação pode estar relacionada, e consequentemente o número de partições deve refletir a intuição coletiva sobre como devem estar organizados os documentos. Assim, propomos que, uma vez executado o algoritmo de deteção de comunidades, o número de comunidades (com mais de um documento) seja k, e as sementes serão os documentos que têm maior grau dentro da sua comunidade (a escolha aleatória é utilizada sempre que se verifique um empate entre os documentos).

3.3.3. Algoritmo k-*Communites* (k-C)

Do exposto anteriormente surge o algoritmo k-*Communities* (k-C) que inicia com k sementes, cada uma coincidente com os vetores dos documentos que obtiveram maior grau dentro da sua comunidade. Para além disso, os novos centróides serão os documentos que são mais similares aos restantes documentos dentro do respetivo *cluster* utilizando para isso a similaridade dos cossenos.

O passo 1 do algoritmo será apenas recalculado quando o sistema identificar a entrada de novas *tags* que justifique nova execução do algoritmo de deteção de comunidades. Os passos do algoritmo são:

ALGORITMO K-COMMUNITIES

1. k corresponde ao número de comunidades (com mais de s documentos) numa rede de tags (onde os documentos são os nós e cada aresta liga documentos que partilham *tags*) : cada semente é o vetor do documento que tem o maior grau dentro da sua comunidade.
2. Calcula a similaridade dos cossenos entre cada documento e todas as sementes.
   (a) Se a similaridade dos cossenos entre um documento e todos os centroides for m então para o cálculo, retorna ao passo 1 e adiciona este documento ao conjunto das sementes.
   (b) Else if gera os *clusters* associando cada documento à sua semente mais próxima.
3. Cálculo do novo centróide de cada *cluster*, escolhendo o documento que é mais similar aos restantes documentos do *cluster*. Assim, a similaridade dos cossenos entre cada documento e todos os outros documentos dos *cluster* é calculada, e o documento escolhido é o que obtém soma máxima como se mostra na Equação 21 (escolha aleatória se ocorrer empate entre documentos).

$$max \sum_{j=1}^{n} \cos(d_i, d_j) \qquad \text{Equação 21}$$

4. Volta ao passo 2. O processo para quando convergir para um mínimo local.

Observe-se o seguinte exemplo ilustrativo da execução do algoritmo:
Na Figura 64 podemos constar que foram criadas 9 comunidades quando executado o algoritmo de deteção de comunidades Girvan-Newman descrito na Secção 2.3.1. Foram consideradas como sementes os documentos: 3; 10; 108; 124; 32; 38; 52; 73 e 75.

Na Figura 65 estão representados os *clusters* formados na 1.ª iteração. A cada documento está associada uma barra cujo comprimento traduz a soma da similaridade dos cossenos aos restantes documentos do respetivo *cluster*. A verde estão sinalizadas as barras das sementes iniciais e a vermelho estão identificadas as barras dos documentos que passaram ser os novos centroides. Note-se que na imagem foram omitidos dois *clusters* por ambos terem apenas dois documentos (que por consequência não implicou alteração dos respetivos centroides).

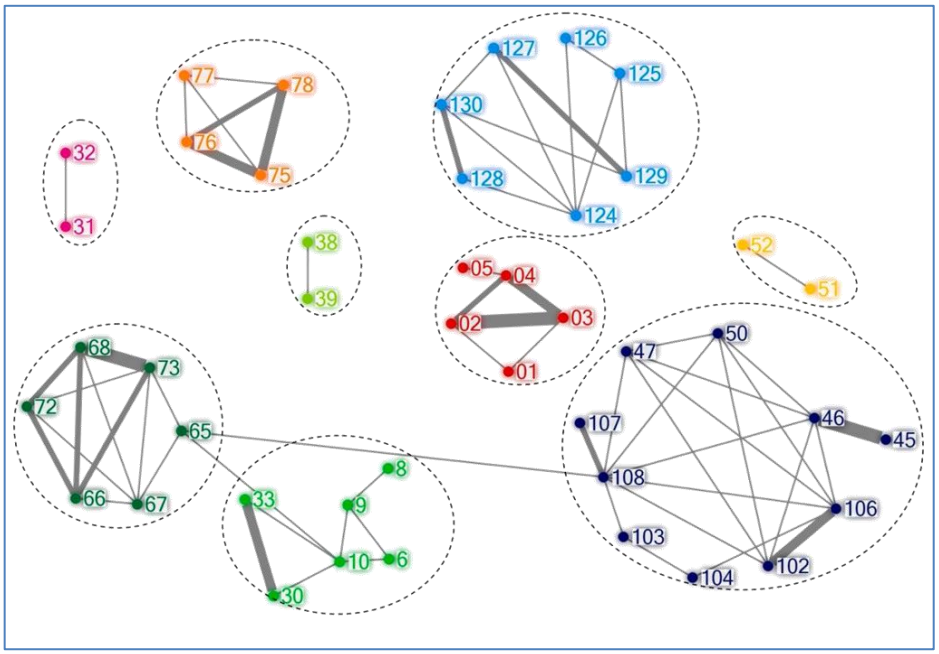


Figura 64: Resultado da execução do algoritmo de deteção de comunidades Girvan -Newman

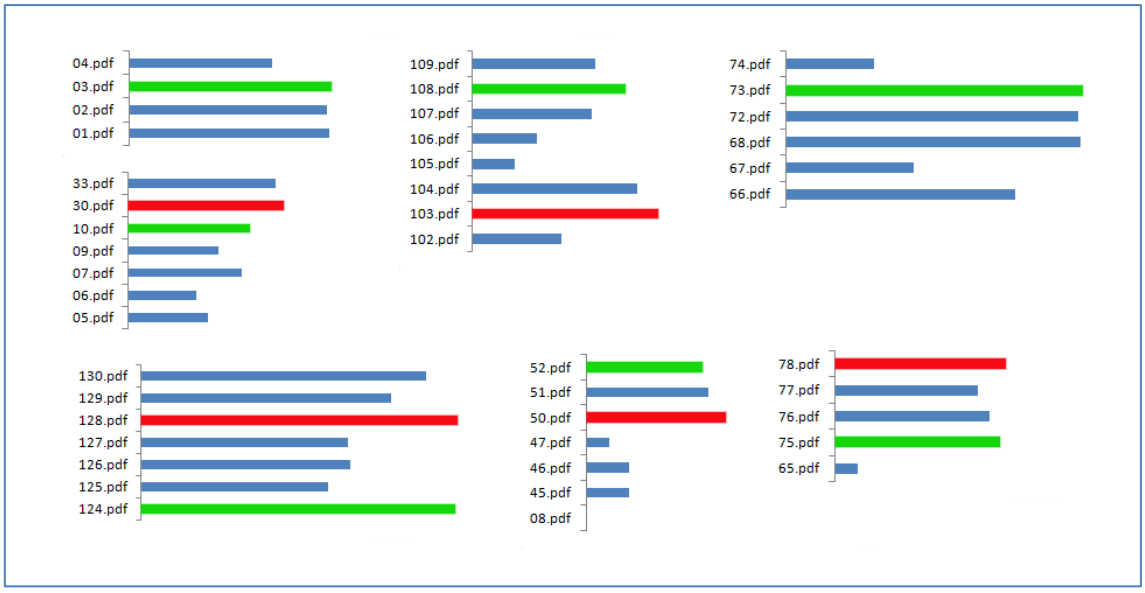


Figura 65: Determinação dos novos centroides para os *clusters* com mais de dois documentos usando o algoritmo k-C.

Os novos centroides são: 3; 30; 103; 128; 32; 38; 50; 73 e 78. Segue-se a associação de cada documento ao seu centróide mais próximo, formando-se assim novos *clusters*. Depois dos *clusters* formados, recalculam-se os centroides. O processo termina quando não houver mais alterações nos *clusters* formados.

3.3.4. Complexidade Temporal

Ainda que com a escolha cuidada das sementes seja esperado que o número de iterações seja menor do que no algoritmo *k-means*, do ponto de vista da complexidade

temporal, e uma vez que é necessário comparar a distância entre todos os documentos dentro do seu *cluster* em cada iteração, há um custo a pagar em termos de complexidade temporal.

A complexidade inerente à associação de cada documento à respetiva semente será tal como no algoritmo *k-means* $O(kn)$, em que n é o número de iterações e k é o número de sementes.

Contudo, no algoritmo k-C, ainda é necessário calcular em cada iteração o centróide mais próximo, exigindo que sejam calculadas todas as distâncias entre todos os documentos para cada um dos k *clusters*. Portanto, o custo em termos de complexidade temporal está relacionado com o tipo de dados. Se, por exemplo, num *cluster* ficarem quase todos os documentos e nos restantes apenas um documento em cada *cluster*, o número de cálculos necessário será dado através da seguinte equação:

$$C_2^{n-(k-1)} = \frac{(n-k+1)(n-k)}{2} \qquad \text{Equação 22}$$

Por outro lado, se supusermos que os documentos ficam uniformemente distribuídos pelos *clusters*, obtemos o menor custo, dado pela equação:

$$k \times C_2^{n/k} = k \times \frac{n^2-kn}{2k^2} \qquad \text{Equação 23}$$

Onde $\frac{n}{k} \in IN \ \wedge \frac{n}{k} > 1$

No gráfico apresentado na Figura 66 apenas é visível a linha do pior caso quando estamos a falar de 7 *clusters* (as restantes linhas do pior caso não são visíveis porque foram sobrepostas por esta última). Isto significa que não há alterações significativas quando é alterado o número de *clusters*.

Por outro lado, quando analisamos o melhor caso podemos constatar que o mesmo não acontece, sendo visível que à medida que aumentamos o número de *clusters* também o custo em termos de complexidade temporal diminui. Sendo ainda de referir que o pior caso apresenta resultados superiores aproximadamente na razão de k, sendo k o número de *clusters*, pois a razão entre o pior caso e o melhor caso é dada pela Equação 24 e o limite quando n tende para +∞ é k.

$$k + \frac{k}{n} - \frac{k^2}{n} \qquad \text{Equação 24}$$

Assim, a título de conclusão, no pior caso a complexidade associada por iteração é:

$$O(kn + ((n-k+1)(n-k))/2) \qquad \text{Equação 25}$$

Ou seja, a complexidade é O(n^2).

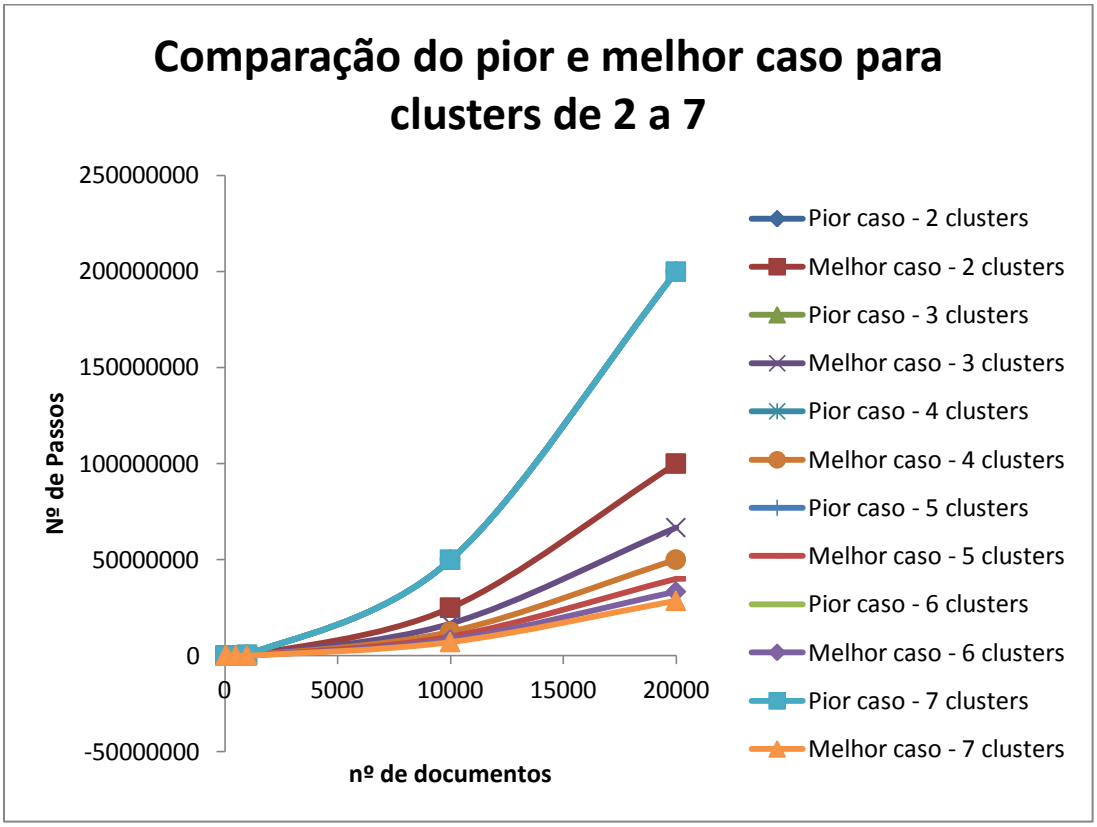


Figura 66: Comparação do custo de execução entre o pior e o melhor caso considerando entre 2 a 7 *clusters*.

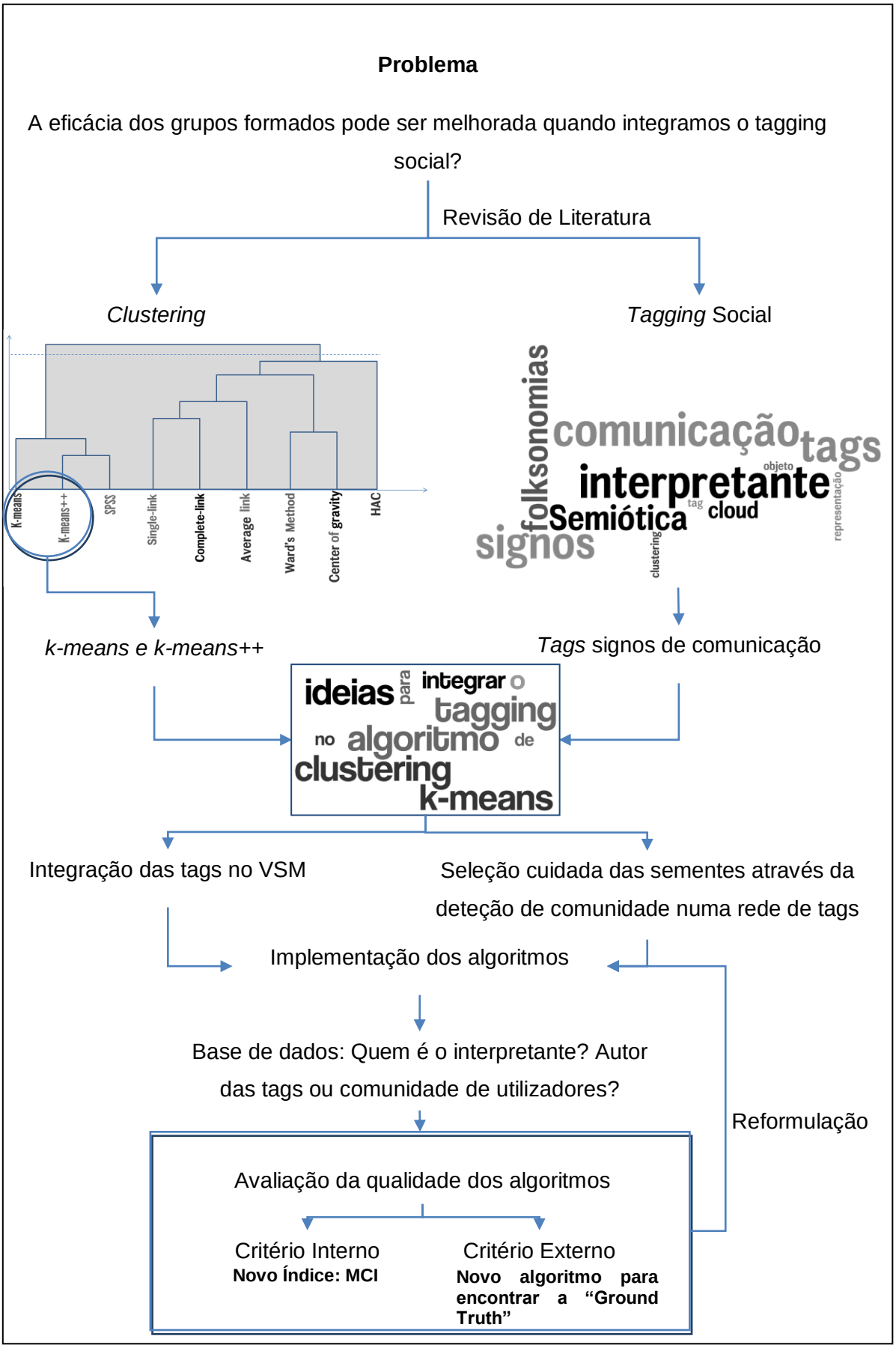
Capítulo 4
Avaliação

4.1. Opções e Procedimentos de Caráter Metodológico

No sentido de apresentar as opções e procedimentos metodológicos elaborámos o Esquema 2. Assim, considerando a problemática de investigação foi feita uma revisão de literatura em duas áreas principais: *Clustering* e *Tagging Social*. Dos vários algoritmos de *clustering* analisados e estudados optámos pelo algoritmo *k-means* que é consensualmente considerado simples e eficiente. Contudo, o facto de ser eficiente, isto é de gerar rapidamente os *clusters*, não garante que seja eficaz, ou seja, que gera agrupamentos que fazem sentido. Visando melhorar a sua eficácia, sugerimos algoritmos alternativos, tendo por base duas abordagens diferentes.

Assim, a primeira abordagem consistiu na integração das *tags* diretamente no *Vector Space Model* (VSM) de acordo com a sua ocorrência no documento, utilizando o algoritmo *k-means* e *k-means*++.

A segunda abordagem consistiu na análise da rede de *tags* para determinar especificamente quais as sementes. Esta nova abordagem dá origem a um algoritmo de *clustering* similar ao algoritmo *k-means*, a que chamámos *k-Communities* (K-C) que inicia, tal como o *k-means*, com k sementes, mas coincidentes com os próprios documentos, obtidos através da rede de *tags*. No algoritmo K-C o novo centróide é o documento que está mais próximo dos restantes documentos de cada *cluster*, usando a similaridade dos cossenos. A desvantagem é que para encontrar este novo centróide é necessário calcular a similaridade dos cossenos entre todos os documentos que fazem parte de cada *cluster* em cada iteração, tornando-o menos eficiente por iteração quando comparado com o algoritmo *k-means*.



Esquema 2: Procedimentos de caráter metodológico

Seguiu-se a implementação dos algoritmos e respetiva avaliação. A seleção dos repositórios teve em consideração 2 cenários: os contextos em que é o autor das *tags* que vai ser o interpretante, ou seja as *tags* são atribuídas por um único utilizador; e repositórios em que a interpretação é feita no contexto de uma comunidade.

A avaliação dos algoritmos pode ser feita recorrendo a medidas de avaliação interna e medidas de avaliação externa. Como estas últimas só podem ser implementadas quando conhecemos a estrutura do repositório foi desenvolvido um algoritmo baseado na informação interna dos documentos, como as descrições providenciadas pelos utilizadores através das *tags* e da distância entre os documentos. Por outro lado, foi criado um novo índice para a avaliação interna, MCI.

4.2. Avaliação do Clustering

Diferentes algoritmos de *clustering* podem produzir diferentes *clusters*. Mesmo a aplicação do mesmo algoritmo de *clustering*, usando os mesmos parâmetros, pode providenciar resultados diferentes (por exemplo, no caso do algoritmo *k-means,* a escolha aleatória das sementes pode gerar *clusters* diferentes em cada execução para o mesmo k). Portanto, determinar a qualidade dos *clusters* formados é um dos maiores desafios no âmbito do problema do *clustering*.

De acordo com a literatura, o resultado dos algoritmos de *clustering* pode ser avaliado por três técnicas diferentes: critério externo; critério interno e critério relativo (Halkidi & Vazirgiannis, 2001; Tasdemir & Merenyi, 2011; Theodoridis & Koutroumbas, 2009).

O critério externo avalia a qualidade do *clustering* comparando o resultado de *clustering* com uma estrutura que tem por base a intuição humana e portanto utiliza informação externa não presente nos dados. Diferentemente, o critério interno avalia o resultado de um algoritmo de *clustering* utilizando apenas informação presente nos dados. Por fim o critério relativo visa encontrar o melhor esquema de agrupamento, comparando os resultados obtidos pelo mesmo algoritmo mas usando diferentes parâmetros. De acordo com Theodoridis e Koutroumbas (2009), o critério relativo não requer testes estatísticos, evitando grandes exigências computacionais.

4.3. Medidas de Avaliação

É um procedimento corrente utilizar medidas de avaliação externas e internas para avaliar a qualidade dos algoritmos. As medidas de avaliação interna medem a separação e a compacidade, isto é, quando os grupos estão bem separados e simultaneamente os

documentos dentro de cada *cluster* estão próximos, indicando qual o algoritmo que tem melhor performance para um determinado conjunto de dados. Assim, comparar estas medidas quando analisamos os resultados de diferentes algoritmos de *clustering* resulta apenas numa aproximação no que diz respeito ao problema de otimização geral (Feldman & Sanger, 2007).

Contudo, estas medidas não garantem a eficácia dos grupos formados, e apesar do julgamento humano ser subjetivo, é também importante considerar as medidas de avaliação externa no sentido de medir o grau de similaridade entre os grupos formados automaticamente e os grupos manuais.

4.3.1. Medidas de Avaliação Interna

Existem várias medidas de avaliação interna presentes na literatura. Estas medidas medem a compacidade e separação variando a forma como são calculadas. Por exemplo, no índice de Dunn (Dunn, 1974), compacidade é calculada usando a raiz quadrada da distância máxima entre quaisquer dois documentos do mesmo *cluster*, portanto usando o diâmetro do *cluster*. Por outro lado, o índice DB (Davies & Bouldin, 1979) mede a compacidade baseado nas similaridades entre cada *cluster* e todos os outros *clusters*, calculando a soma da dispersão de dois *clusters*.

Consideramos que, mais do que a distância entre os documentos em cada *cluster*, é importante identificar se cada documento e o seu documento mais próximo fazem parte do mesmo *cluster*. Portanto, a nossa proposta passa por medir a compacidade de acordo com este princípio, e a separação através da distância ao *cluster* mais próximo, construindo em termos da separação a medida sobre o pior caso, tal como acontece no DB *Index* (Dunn, 1974).

a. Maximum Cosine Index

Para construir o novo índice, a que vamos chamar *Maximum Cosine Index* (MCI), precisamos de considerar uma rede de documentos onde cada documento está conectado ao seu documento mais próximo (usando a similaridade dos cossenos). Pretende-se medir a distância entre cada *cluster* e o seu *cluster* mais próximo e calcular quantas vezes é superior à média das distâncias entre cada documento e o seu documento mais próximo dentro de cada *cluster*.

Na Figura 67 podemos ver os documentos que pertencem a cada *cluster*. As linhas a tracejado indicam a distância entre cada *cluster* e o *cluster* mais próximo e as linhas sólidas indicam a distância de cada documento ao documento mais próximo.

Em cada *cluster* estão identificados os diferentes componentes conexos com cores diferentes. As distâncias que aparecem em cada *cluster* serão pesadas de acordo com o seu componente conexo (as distâncias que aparecem em cada componente conexo terão peso igual ao número documentos presentes nesse componente conexo). Por exemplo, na Figura 67, no canto superior esquerdo, as distâncias dos documentos 102, 103, 104, 105, 106 e 109 ao documento mais próximo terão peso 6 porque neste componente conexo (identificado com a cor azul) temos 6 documentos.

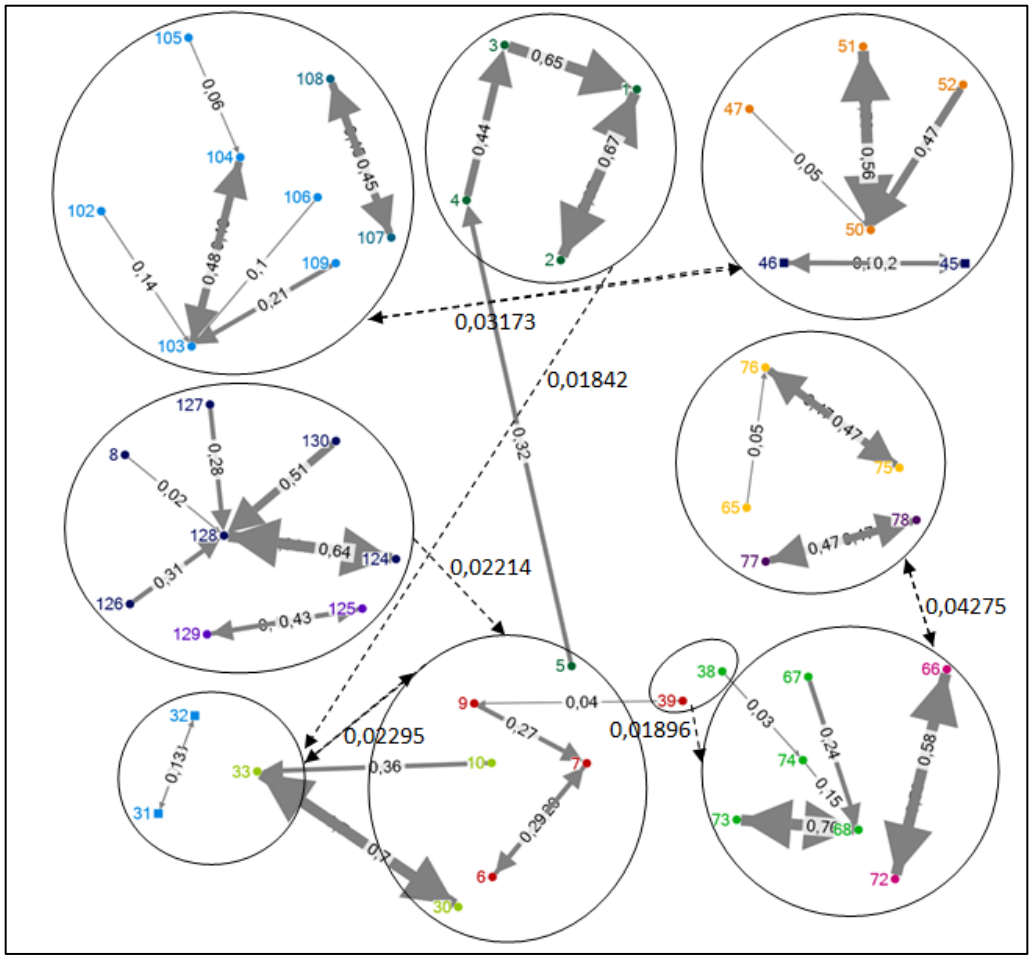


Figura 67: Representação da distância do documento mais próximo de cada documento (linhas sólidas) e da distância entre cada *cluster* e o *cluster* mais próximo (linhas a tracejado) (Cunha, Figueira, & Mealha, 2013a)

A proporção entre a compacidade e a separação dos *clusters* (com mais de um documento) é dada pela Equação 26.

$$X_i = \frac{R_i}{d_i} \qquad \text{Equação 26}$$

Onde:

R_i – média ponderada das "distâncias" ao documento mais próximo dentro de cada *cluster*.

d_i – distância ao *cluster* mais próximo.

De seguida determinamos a média ponderada dos vários *clusters*. O peso atribuído a cada *cluster* será igual ao número de documentos que tem o documento mais próximo dentro do *cluster* como mostra a Equação 27.

$$\mathrm{MCI} = \frac{\sum_{i=1}^{n} x_i p_i}{\sum_{i=1}^{n} p_i} \qquad \text{Equação 27}$$

Onde:

$$p_i = \frac{s_i}{t_i}$$

s_i – Número total de documentos do *cluster i* que tem o documento mais próximo dentro do *cluster*.

t_i – número total de documentos do *cluster i*

Para o exemplo apresentado na Figura 67, temos que, em média, a "distância" entre cada *cluster* e o *cluster* mais próximo é 13,8 vezes superior à média das "distâncias" entre cada documento e o documento mais próximo dentro de cada *cluster*.

4.3.2. Medidas de Avaliação Externa

A utilização de medidas de avaliação externa para avaliar a qualidade de um algoritmo pressupõe a comparação de duas estruturas: a que resulta da implementação do algoritmo de *clustering* e a que é obtida pressupondo a intuição humana. Várias medidas de avaliação externa têm sido propostas (Manning, et al., 2009), tais como, *Purity*, *Precision*, *Recall*, *F measure* e *Rand Index*. Algumas medidas externas são baseadas na comparação direta entre os *clusters* manuais e os *clusters* automáticos, como é o caso da medida *Purity*, enquanto outras medidas são baseadas nas relações existentes numa coleção de n documentos entre os $n(n-1)/2$ pares de documentos, tais como a medidas F1, *Precision*, *Recall* e *Rand Index*. Portanto, para calcular estas medidas é necessário conhecer as várias relações possíveis entre os pares de documentos (Figura 68): *True Positives* (TP); *True Negatives* (TN); *False Positives* (FP); *False Negatives* (FN).

True Positives (TP): Número de pares de documentos que estão corretamente colocados num *cluster*.

True Negatives (TN): Número de pares de documentos que foram corretamente rejeitadas para a mesmo *cluster*.

False Positives (FP): Número de pares de documentos que estão incorretamente colocados no mesmo *cluster*.

False Negatives (FN): Número de pares de documentos que estão incorretamente colocados em *clusters* diferentes.

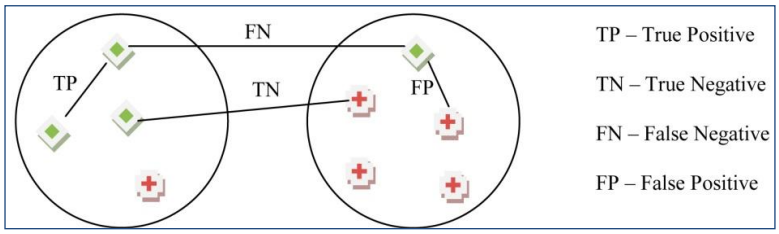


Figura 68: Exemplos de tipos de relações entre pares de documentos (Cunha & Figueira, 2012).

a. Purity

A medida *Purity* (Feldman & Sanger, 2007) compara as classes manuais com os *clusters*, selecionando para cada classe o *cluster* mais similar. A percentagem de documentos comuns é dada pela Equação 28.

$$Purity(C, L) = \frac{1}{n}\sum_k max_j \left|C_k \cap L_j\right| \qquad \text{Equação 28}$$

Onde $C = \{C1, C2, \dots, Cm\}$ é o conjunto dos *clusters* e $L = \{L1, L2, \dots, Lm\}$ é o conjunto das classes.

Um mau *clustering* tem *Purity* próximo de zero e um *clustering* perfeito tem *Purity* 1. De qualquer modo, é necessário ter em atenção que quando o número de *clusters* é grande, é fácil obter um valor próximo de 1 (aliás no caso limite de cada documento corresponder um *cluster*, *Purity* será 1).

b. Precision

A medida *Precision* (Manning, et al., 2009) indica a percentagem de pares de documentos que estão corretamente colocados no mesmo *cluster*.

$$P = \frac{TP}{TP+FP} \qquad \text{Equação 29}$$

c. Recall

Recall indica a proporção entre os pares de documentos que estão corretamente colocados no mesmo *cluster* e os pares de documentos que estão no mesmo *cluster* conjuntamente com os que deviam estar no mesmo *cluster*.

$$R = \frac{TP}{TP+FN}$$ Equação 30

d. F *Measure*

F *Measure* corresponde à média harmónica ponderada entre o *Precision* e o *Recall*.

$$F_\beta = \frac{(\beta^2+1)\times P\times R}{\beta^2 P+R}$$ Equação 31

Assim, F1, indica a média harmónica entre o *Precision* e o *Recall* tendo ambos o mesmo peso e é calculada como mostra a Equação 32.

$$F_1 = \frac{2\times P\times R}{P+R}$$ Equação 32

Se considerarmos $\beta > 1$ vamos enfatizar o *Recall* e se optarmos por $\beta < 1$ vamos estar a enfatizar o *Precision*.

e. Rand Index

Rand Index calcula a percentagem de decisões corretas, isto é, pares de documentos colocados no mesmo *cluster* e os pares de documentos que estão corretamente colocados em *clusters* diferentes.

$$RI = \frac{TP+TN}{TP+TN+FN+FP} = \frac{2\times(TP+TN)}{n^2-n}$$ Equação 33

f. Reflexão Sobre as Medidas Externas

Todas estas medidas dão informações relevantes. Contudo, um valor alto no *Rand Index* não significa que a medida F1 indique igualmente uma alta percentagem uma vez que o cálculo do *Rand Index* é feito sobre a totalidade de pares possíveis.

Nesse sentido, é natural que o número de pares verdadeiros negativos seja muito superior às restantes possibilidades e, se o número de verdadeiras positivas for próximo do número de pares Falsos Positivos e Falsos Negativos, vamos garantidamente obter uma percentagem superior de *Rand Index* do que da medida F1 pois esta, ao estar dependente das medidas *Precision* e *Recall*, também está diretamente dependente dos erros considerados em cada um destes cálculos, ou seja das Falsas Negativas e das Falsas Positivas.

Assim, é importante perceber se o algoritmo se revela eficaz em não juntar nos mesmos *clusters* documentos que não devem pertencer aos mesmos *clusters*. Esta informação é-nos dada pela medida *Rand Index*.

Por outro lado, é importante conhecer a influência das Falsas Positivas e das Falsas Negativas para o cálculo do F1, pois quanto menor for o número de Falsas positivas e de Falsas negativas maior será a medida F1.
Apesar de ignorar documentos que podem estar corretamente associados a um *cluster*, a medida *Purity* permite-nos ter a perceção do número de documentos que em maior número continuam a pertencer ao mesmo *cluster*, sendo que, à medida que se aproxima de 1, também os *clusters* automáticos se aproximam das classes manuais.

4.3.3. Método para Obtenção da *"Ground Truth"*

A implementação das medidas de avaliação externa descritas no Capítulo 4.3.2 pode não ser possível, especialmente se não conhecermos a estrutura do repositório ou no caso do repositório ser demasiado grande para ser organizado manualmente pelo Homem. Para além disso, a classificação de documentos é subjetiva porque há mais do que uma maneira de classificar corretamente os documentos. Contudo, seria desejável que a organização manual dos documentos refletisse a nossa intuição coletiva sobre a estrutura do *clustering*.

De acordo com Levy (1997) todos podem contribuir para a construção do conhecimento, melhorando a construção da inteligência coletiva. Portanto, porque não utilizar as anotações feitas pelos utilizadores de uma comunidade para avaliar a qualidade dos algoritmos de *clustering*? Assim, as *tags* vistas como signos de comunicação, podem ser interpretadas na perspetiva da comunidade de utilizadores, no sentido de relacionar a informação presente através da deteção de padrões.

Assim, descreveremos como podemos utilizar as *tags* e a similaridade entre os documentos para obter uma "*Ground Truth* Automática".

a. Metodologia para Encontrar a "*Ground Truth* Automática"

Propomos um método para encontrar as classes que podem substituir os grupos manualmente criados, baseado em três passos: **(a)** aplicação de um algoritmo de deteção de comunidades numa rede de documentos, onde a relação binária é estabelecida entre pares de documentos que partilham pelo menos uma *tag* – cada comunidade corresponde a uma classe da "*Ground Truth* Automática"; **(b)** no sentido de melhorar a qualidade das classes encontradas no passo (a) vamos utilizar a informação interna baseada na seguinte ideia: o documento mais próximo de cada documento estará geralmente no mesmo *cluster*. A similaridade dos cossenos foi a nossa escolha para encontrar o documento mais similar de cada documento porque não depende do

comprimento do vetor do documento. Portanto, neste passo, vamos considerar outra relação binária: d_i está conectado ao documento d_j se o documento d_j é o documento mais próximo do documento d_i. No passo **(c)** é feita a integração dos passos (a) e (b), combinando a informação do grafo das *tags* (onde a estrutura da comunidade foi identificada) e do grafos das distâncias, permitindo a fusão ou separação das classes encontradas no passo (a).

b. Notas Sobre o Documento Mais Próximo

Durante o processo de *clustering*, esperamos que cada documento seja colocado no mesmo *cluster* do seu documento mais próximo. Contudo, isto pode não ser confirmado, particularmente se os documentos estiverem muito afastados.

No grafo dirigido apresentado na Figura 69, cada nó representa um documento e cada aresta a conexão ao documento mais próximo, nela está indicada a similaridade dos cossenos entre os documentos. Este peso é visualmente indicado pela grossura das arestas. Olhando para a esquerda da Figura 69, podemos ver que o documento mais próximo do documento 38 é o documento 74. Contudo, a similaridade dos cossenos entre os dois documentos é aproximadamente 0,04. Dito de outro modo, o ângulo entre os dois documentos é aproximadamente 88º, significa que os documentos estão bastante afastados.

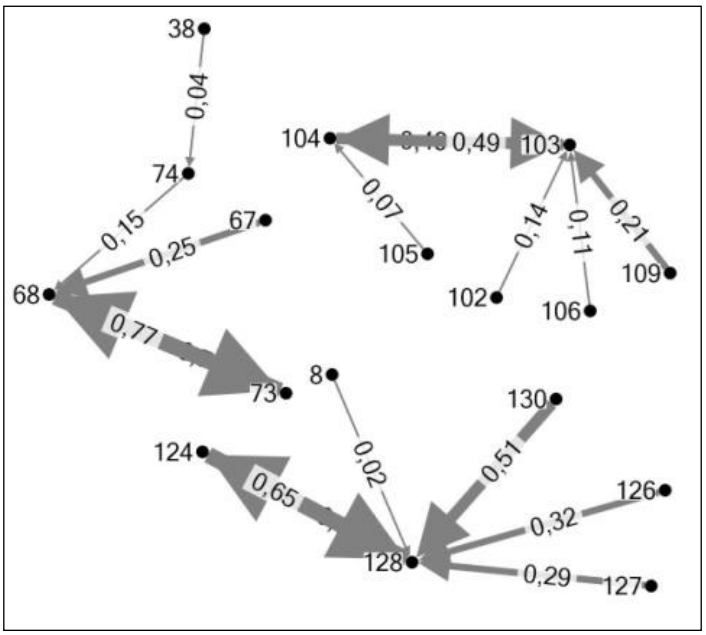


Figura 69: Parte de um grafo dirigido, onde cada documento está conectado ao seu documento mais próximo usado a similaridade dos cossenos (Cunha & Figueira, 2012).

Ainda assim, não podemos afirmar que documentos distantes não estão conectados porque documentos relacionados não precisam de estar necessariamente próximos. Dois documentos podem dizer respeito ao mesmo tópico mas as palavras utilizadas podem ser muito diferentes em cada um, ou podem fazer parte da mesma área do conhecimento mas não tratar do mesmo assunto. Por exemplo, na Figura 69, os documentos 102, 103, 104, 105, 106 e 109 estão relacionados porque são todos do campo do conhecimento da Biologia (estamos certos disto porque são artigos da secção da Biologia do Repositório aberto da Universidade do Porto). Neste caso, apesar de ser expectável que o documento mais próximo de cada documento estivesse próximo, estão em geral distantes como se pode ver pela grossura das linhas. Portanto, cruzando esta informação com as conexões entre os documentos, através das *tags*, é crucial decidir quando os documentos podem ou não fazer parte do mesmo *cluster*.

Deste modo, e uma vez que a separação dos documentos em *clusters* está diretamente dependente do repositório, consideramos a distância entre cada documento e o seu documento mais próximo e o tipo de distâncias observadas. Por exemplo, podemos ter um repositório onde a distância entre cada documento e o seu documento mais próximo varia entre 0,5 e 1 e outro repositório onde este intervalo pode variar entre 0,1 e 0,5. No primeiro cenário, documentos cuja distância entre eles é de 0,5 estão muito afastados, ao contrário do segundo cenário, onde a distância entre os documentos indica que são os mais próximos daquele repositório.

No sentido de analisar os resultados das similaridades dos cossenos observadas entre cada documento e o documento mais próximo, sugerimos o cálculo do mínimo, do primeiro quartil e da mediana. Se a similadidade dos cossenos entre os documentos é inferior ao 1.º quartil, então apenas 25% estão nestas circunstâncias, indicando uma conexão fraca entre os documentos.

Por outro lado, se a similaridade dos cossenos entre os documentos for superior à mediana, corresponderá a 50% dos resultados observados.

É por isso que para a deteção automática da "*Ground Truth*" vamos considerar os quartis: quartil 1 (Q1) e mediana (Q2).

c. Algoritmo da "*Ground Truth* Automática"

No passo (a) selecionamos um algoritmo de deteção de comunidades. Para os exemplos apresentados nesta secção escolhemos o algoritmo Girvan e Newman (Girvan & Newman, 2002), considerado por Fortunato (2009), um marco na campo da deteção de

comunidades. Contudo, durante a execução dos testes, para avaliar os algoritmos de *clustering* propostos, outros algoritmos foram selecionados, especialmente quando o tamanho do repositório exigir algoritmos mais escaláveis.

No sentido de integrar os passos (a) e (b) descritos na secção 4.3.3.a vamos considerar o seguinte algoritmo:

Seja $G^T=(V,E^T)$ o grafo das *tags* e $G^D=(V,E^D,W^D)$ o grafo dirigido das distâncias. V é um conjunto finito de vértices, no qual cada vértice é um documento; e E é o conjunto das conexões (arestas) entre os vértices.

G^D é um grafo pesado e (v_i,v_j) é pesado através do cosseno do ângulo entre dois documentos (vértices). Assim $\cos(90) \leq W_{ij}^D \leq \cos(0)$.

1. Para cada aresta dirigida (v_i,v_j) presente no grafo das distancias G^D:
   (a) Se v_i e v_j fazem parte da mesma comunidade em G^T vai para o passo 1.
   (b) Se v_i e v_j não fazem parte da mesma comunidade em G^T e o nó vi tem grau zero, então transfere o documento vi para a comunidade do documento vj; caso contrário:
      (i) $W_{ij}^D \leq Q_1$ (*weak connection*) vai para o passo 1
      (ii) $Q_1 < W_{ij}^D \leq Q_2$ (*questionable connection*)
      - Se a aresta dirigida (v_j,v_i) está presente no grafo das distâncias G^D, transfere para a comunidade do documento que tem o menor grau dentro da sua comunidade em G^T e vai para o passo 1 (*mutually closest document*).
      - Se a aresta dirigida (v_j,v_i) não está presente no grafo das distâncias G^D, vai para o passo 1.

      (iii) $W_{ij}^D \geq Q_2$ (*strong connection*)
      - Se a aresta dirigida (v_j,v_i) está presente no grafo das distâncias G^D, transfere para a outra comunidade o documento que tem menor grau dentro da sua comunidade, caso contrário.
      - Transfere o documento v_j para a comunidade do documento v_i se $j<i$ ou transfere v_i para a comunidade do documento v_j se $j>i$ e vai para o passo 1.

O algoritmo para quando não ocorrerem mudanças de comunidade.

De seguida, vamos ilustrar a execução de alguns dos passos do algoritmo. Na Figura 70, temos que o documento mais próximo do documento 8 é o documento 128, mas a distância é inferior ao Q1 (quartil 1) e por isso os documentos não alteram de comunidade.

Na Figura 71 temos dois documentos, cuja conexão é questionável mas como estão mutuamente próximos a decisão é transferir o documento 7 para a comunidade do documento 6 porque o grau do documento 6 é maior do que o grau do documento 7 dentro das respetivas comunidades.

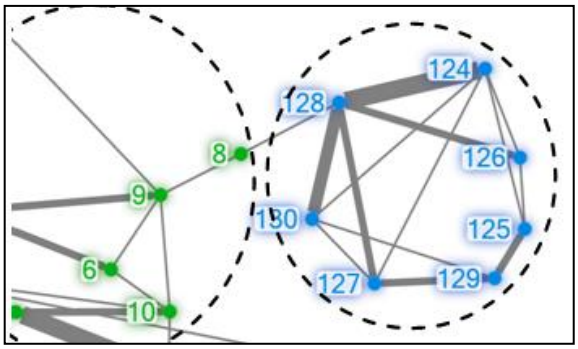


Figura 70: $(v_8, v_{128}) \in G^D$ e $w^D_{8,128} \leq Q_1$ – *weak connection* (Cunha & Figueira, 2012).

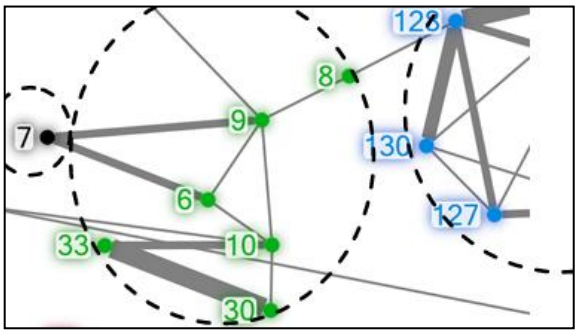


Figura 71: $(v_7, v_6) \in G^D$ e $(v_6, v_7) \in G^D$ (*mutually closest document*) e $Q_1 \leq w^D_{7,6} \leq Q_2$ (*questionable connection*) (Cunha & Figueira, 2012).

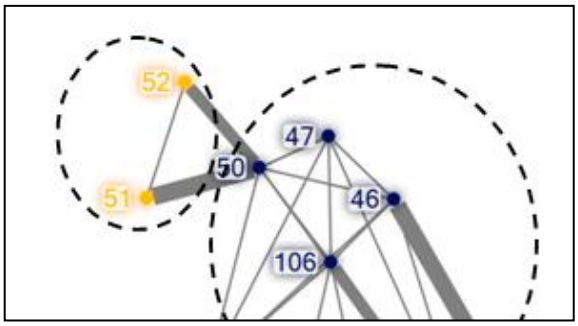


Figura 72: $(v_{51}, v_{50}) \in G^D$; $(v_{50}, v_{51}) \in G^D$ $w^D_{51,50} \geq Q_2$ (*strong connection*). $(v_{52}, v_{50}) \in G^D$; $(v_{50}, v_{52}) \notin G^D$ $w^D_{52,50} \geq Q_2$ (*strong connection*) (Cunha & Figueira, 2012).

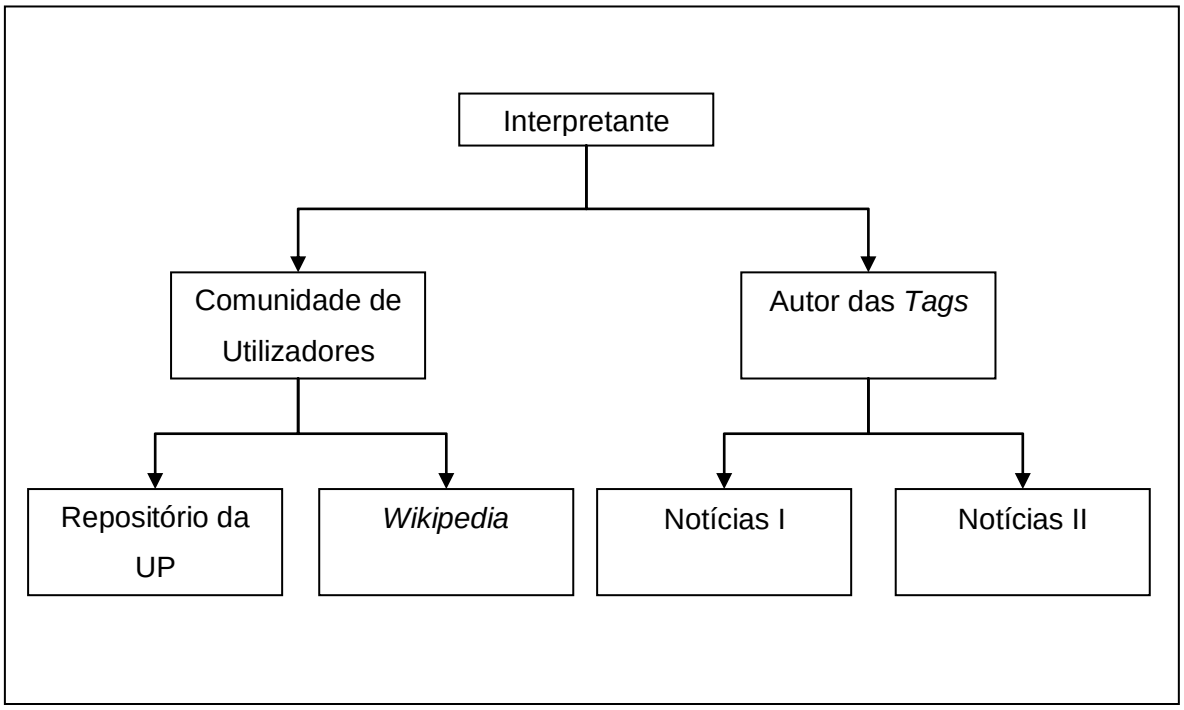
Finalmente, o ultimo exemplo mostra que o documento 51 e o documento 50 estão mutuamente próximos e têm uma ligação forte. O documento 51 é transferido para a comunidade do documento 50, porque este último é o que tem maior grau dentro da sua comunidade. Por outro lado, o documento 52 também tem como documento mais próximo do documento 50 mas não estão mutuamente próximos, a decisão é de transferir o documento 52 para a comunidade do documento 50.

4.4. Roteiro dos Testes a Serem Realizados

A seleção dos repositórios teve em consideração se o interpretante é o utilizador de uma comunidade ou o autor das *tags*. Assim, como se pode ver no Esquema 3, selecionámos dois repositórios em que o interpretante é a comunidade de utiizadores: o primeiro contém 142 artigos científicos do Repositório aberto da Universidade do Porto e o segundo contém 1170 documentos da *Wikipedia*. Quando o interpretante é o autor das *tag*, selecionámos dois repositórios de notícias: o primeiro contém 124 notícias e o segundo contém 65 notícias.

Em todos os repositórios é feito um pré-tratamento dos dados onde:

- Removemos as *stop words;* a pontuação; espaços em branco; e números*;*
- É aplicado o método $Tf - Idf$.



Esquema 3: Repositórios utilizados nos testes dependendo do interpretante.

A ordem pela qual aparecem os casos de estudo segue a ordem natural desta investigação:

- Caso de estudo I – Repositório da UP
- Caso de estudo II – Repositório de notícias I
- Caso de estudo III – Repositório de notícias II
- Caso de estudo IV – Repositório *Wikipedia*

4.5. Caso de Estudo I – Repositório da Universidade do Porto – interpretante é a comunidade de utilizadores

Neste teste tencionamos avaliar a correlação entra a partição manual e a partição automática obtida através do algoritmo da "*Ground Truth* Automática". Para além disso, pretendemos analisar o impacto da integração das *tags* no VSM usando o algoritmo *k-means*++ e o algoritmo k-C.

4.5.1. Descrição do Repositório

Este repositório contém 142 artigos científicos recolhidos da nossa biblioteca pessoal e do Repositório aberto da Universidade do Porto. O Repositório é organizado hierarquicamente em Comunidades e Coleções. As Comunidades correspondem às faculdades e as Coleções agrupam a produção intelectual de cada comunidade, organizado por tipologia de documentos. Assim, escolhemos algumas faculdades e de seguida selecionamos artigos de áreas específicas.

Tabela 21: Lista das palavras-chave que mais coocorrem entre os artigos

	D1					
	1 ao 10	**30 ao 33**	**38 ao 39 124 ao 130**	**45 ao 52**	**65 ao 78**	**102 ao 109**
Top Tags	analysis Cluster clustering complexity folksonomy k-means search social Tag tagging	alpha cronbach knowledge management	algebra angle coalgebras euclidean geometry hopf measure ring trigonometry wilhelmy	Age enclosure Middle Militar Orders portugal	activity analysis children energy fluctuation intensity obese physical rate stroke swimming velocity	aphanopus aveiro carbo eggs estuary flatfish host intensity portugal prevalence
	D2					
	11 ao 20	**34 to 37**	**53 to 58**	**79 to 89**	**110 to 117**	**131 to 140**
Top Tags	algorithm cluster clustering indices tagging validity	cross validation	children health obesity physical smoking stature	activity adolescents children fitness girls obesity physical cardiorespiratory	atlantic bucephalus molecular morphology parasites portugal	algebra derivation hopf module ring semiperfect
	D3					
	21 to 29	**40 to 44**	**59 to 64**	**90 to 101**	**118 to 123**	**141 to 146**
Top Tags	algorithms cluster clustering K-means measures similarity	fluency hedonic marking processing usability	antimicrobial food resistance salmonella	activity adolescent cardiovascular children exercise fitness obesity overweight physical	rdna portuguese parasite coast	dimension goldie module ring semilocal

Os 142 artigos foram separados em 3 repositórios D_1, D_2 e D_3. Para além disso, consideramos mais 3 repositórios DA_1, DA_2 e DA_3 contendo apenas os resumos dos artigos. Cada repositório contém artigos de 6 diferentes áreas do conhecimento (como se pode ver na Tabela 21). Consideramos ainda como *tags* as palavras-chave atribuídas pelos autores dos artigos.

4.5.2. Considerações Sobre a "*Ground Truth* Automática"

O grafo das *tags* G^T e o grafo das distâncias G^D estão apresentados na Figura 73 e na Figura 74, respetivamente. Para isso utilizámos o *software NodeXl*[7] e o algoritmo de deteção de comunidades Girvan e Newman (Girvan & Newman, 2002).

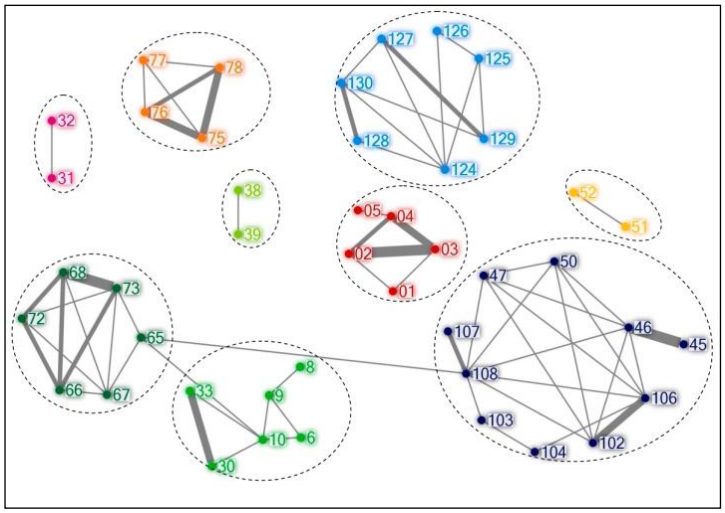


Figura 73: Representação de G^T (Cunha & Figueira, 2012).

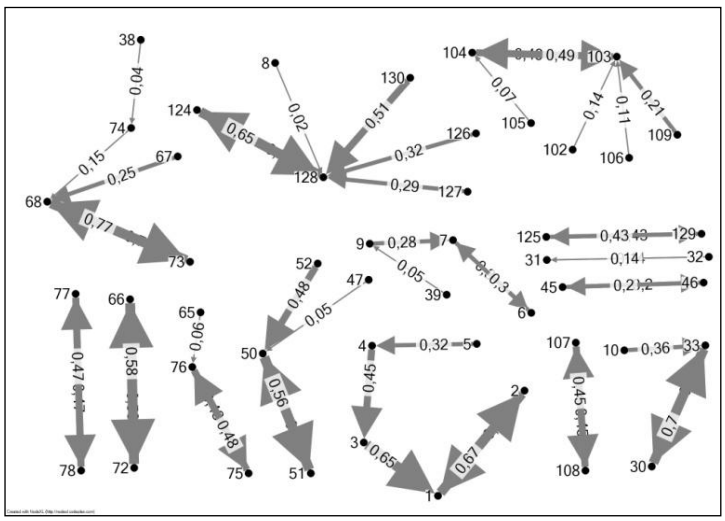


Figura 74: Representação de G^D. Cada aresta é pesada tendo em conta a similaridade dos cossenos entre os documentos (Cunha & Figueira, 2012).

[7] http://nodexl.codeplex.com/

No sentido de facilitar a visualização da implementação do algoritmo, os grafos G^T e G^D foram fundidos, obtendo-se o grafo G^{TUD} como mostra a Figura 75.

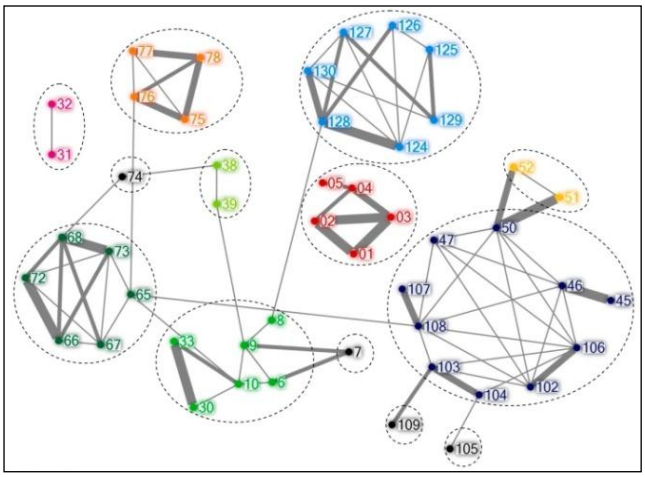


Figura 75: Representação de G^{TUD} no dataset D_1 (Cunha & Figueira, 2012).

Comparando G^T com G^{TUD} observamos que surgiram novos documentos (7, 74, 105 e 109), os que não tinham *tags* em comum com nenhum outro documento. Para além disso, surgem novas arestas, tais como a aresta (v_{52},v_{50}) e a aresta (v_{51},v_{50}). Adicionalmente também podemos ver que G^D tem influência sobre os documentos que já estavam conectados no grafo G^T e que agora têm arestas mais espessas, como é o caso das arestas (v_{107},v_{108}) ou (v_{124},v_{128}).

A obtenção da "*Ground Truth* Automática" para cada repositório (D_1, D_2 , D_3, DA_1, DA_2 e DA_3) coloca na mesma classe (comunidade) documentos provenientes de diferentes áreas do conhecimento. Por exemplo, no repositório D_2, os documentos entre 53 e 58 (área da saúde) e os documentos entre o 79 e o 89 (área das ciências do desporto) são colocados na mesma comunidade depois de executado o algoritmo da "*Ground Truth* Automática". Na Tabela 21 podemos verificar que partilham *tags*, tais como "children", "obesity" e "physical".

Na Figura 76, podemos observar que existe uma conexão entre os artigos destas duas áreas quer pelo número de conexões quer pela espessura das arestas. Isto sugere que é possível encontrarmos conexões semânticas entre documentos relacionados mesmo que provenientes de diferentes áreas do conhecimento.

Uma outra classe de artigos foi detetada pela "*Ground Truth* Automática" e também tem artigos de diferentes áreas do conhecimento (Figura 77). Os documentos pertencem ao repositório D_1 e, de acordo com a Tabela 21, as *tags* presentes sugerem que são

provenientes das áreas História e Biologia. Numa análise mais detalhada a esta lista de *tags* que coocorrem em cada classe, a *tag* "Portugal" é a única comum. A natureza do *tagging* permitiu a reunião destas duas áreas do conhecimento que, poderá ser legitima, caso seja considerado o agrupamento dos artigos por países. Contudo, não valida a relação semântica entre os artigos destas duas áreas.

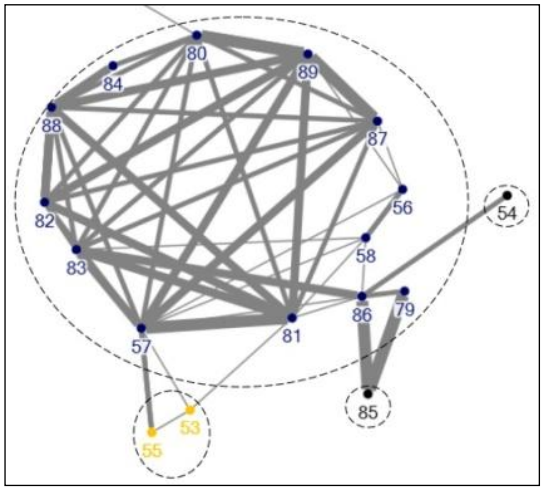


Figura 76: Parte do grafo da fusão entre os grafos G^D e G^T do repositório D_2 (Cunha & Figueira, 2012).

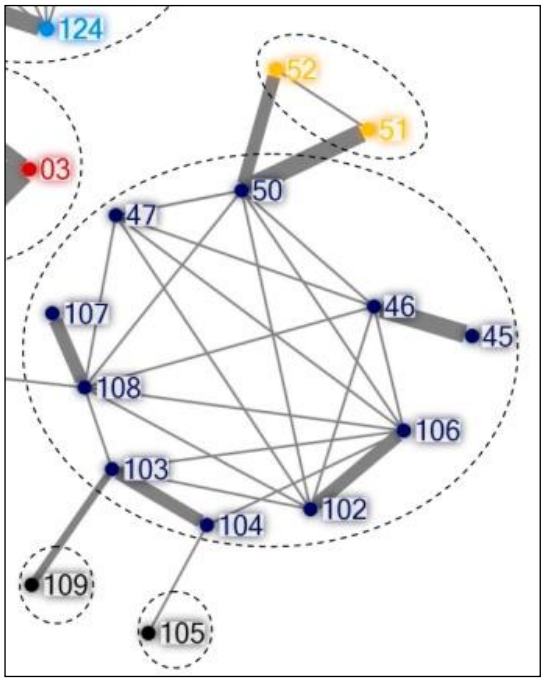


Figura 77: Parte do grafo da fusão entre os grafos G^D e G^T do repositório D_1 (Cunha & Figueira, 2012)

É importante relembrar que, para este estudo, as *tags* usadas foram sugeridas pelos autores de cada artigo. A nossa intuição sugere que a num sistema real de atribuição de *tags*, a *tag* Portugal dificilmente seria das *tags* mais associadas, porque acreditamos que existiriam mais tags consensuais para caraterizar o assunto principal do artigo. Portanto,

é importante que nem todas as *tags* sejam usadas mas apenas as que geram maior consenso entre os utilizadores de uma comunidade. Por outro lado também podemos ver na Figura 77 que a maioria das arestas tem uma espessura fina, revelando a fragilidade das relações entre os documentos.

4.5.3. Comparação da "*Ground Truth* Automática" com os Grupos Manuais

Apesar de não estarmos à espera que as classes detetadas pela "*Ground Truth* Automática" sejam exatamente as mesmas das classes feitas manualmente esperamos que no contexto deste repositório exista uma forte correlação entre elas.

Na Figura 78, apresentamos os resultados das medidas de avaliação externa para os repositórios que utilizam todo o texto dos artigos (D_1, D_2 e D_3) e na Figura 79 apresentamos os resultados para os repositórios que utilizam apenas os resumos dos documentos (DA_1, DA_2 e DA_3). Em cada uma das figuras, a correlação entre os dois gráficos é facilmente percebida. Contudo, é necessário estudar do pondo de vista estatístico se esta correlação é estatisticamente significativa.

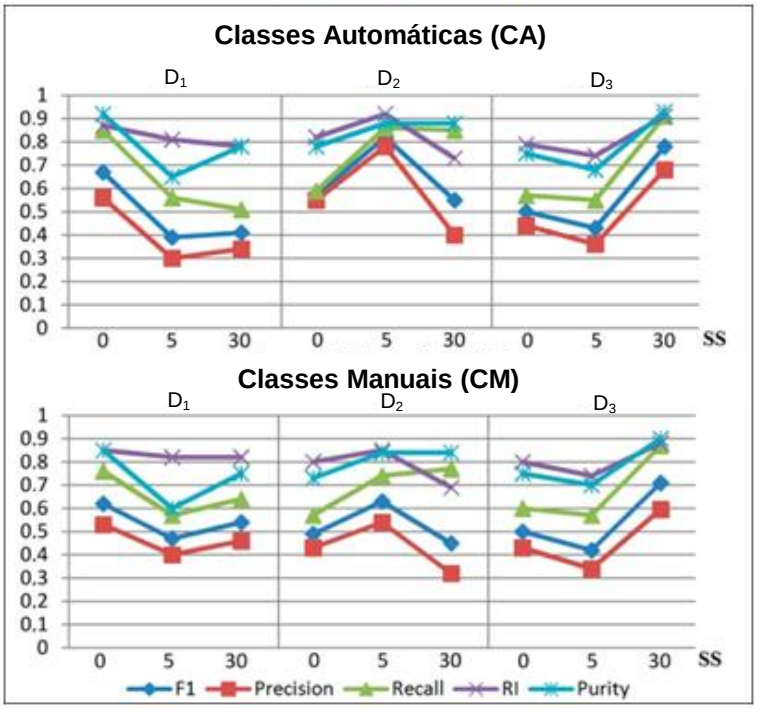


Figura 78: Avaliação externa usando classes automáticas e classes manuais para os repositórios D_1, D_2 e D_3 (Cunha & Figueira, 2012).

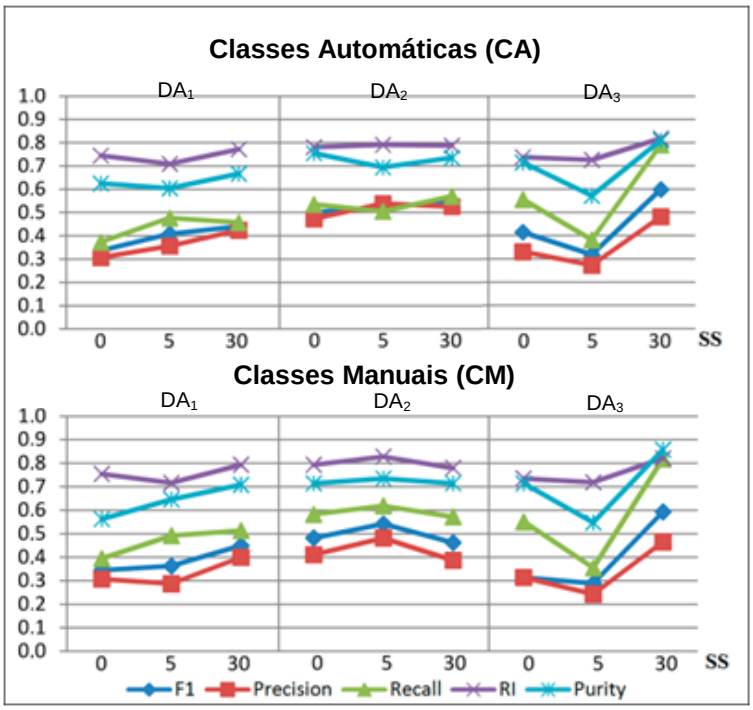


Figura 79: Avaliação externa usando classes automáticas e classes manuais para os repositórios DA_1, DA_2 e DA_3 (Cunha & Figueira, 2012).

Para analisar se efetivamente existe correlação entre os resultados obtidos quando se utilizam classes manuais e automáticas, optámos por analisar os dados através da correlação de Spearman, uma vez que não exige que os resultados sejam normalmente distribuídos. Antes da sua utilização foi necessário verificar se os dados obtidos respeitam os dois princípios exigidos para a sua aplicação, ou seja, uma das exigências está relacionada com o tipo de variável, e neste caso as variáveis que podem ser medidas num intervalo estão incluídas, e existe uma relação de monoticidade entre as duas variáveis (verificado previamente através da análise dos gráficos de dispersão).

Para cada uma das medidas de avaliação F1, *Precision*, *Recall*, *Rand Index* e *Purity* calculamos o coeficiente de correlação quando eram utilizadas classes manuais e classes automáticas.

Os resultados apresentados abaixo indicam que o coeficiente de correlação de Spearman é de 0,561 para a medida F1 mas este resultado não é estatisticamente significante, uma vez que p=0,116 é superior a 0,05 (Tabela 22). Contudo, a medida F1 é obtida através da média harmónica entre o *Precision* e o *Recall* e ambas apresentam correlações estatisticamente significantes. Na Tabela 23, podemos observar que o coeficiente de correlação de Spearman é de 0,711, indicando que existe uma forte correlação positiva entre os resultados da medida *Precision* quando se utilizam Classes Manuais (CM) e

Classes Automáticas (CA), sendo esta correlação estatisticamente significante (p=0,032). O mesmo acontece para a medida *Recall*, sendo neste caso o coeficiente de correlação r_s=0,723 e p=0,028 (Tabela 24).

Tabela 22: Resultado do coeficiente de correlação de Spearman para a medida F1 usando Classes Manuais (CM) e Classes Automáticas (CA).

Correlations

			CM_F1	CA_F1
Spearman's rho	CM_F1	Correlation Coefficient	1,000	,561
		Sig. (2-tailed)	.	,116
		N	9	9
	CA_F1	Correlation Coefficient	,561	1,000
		Sig. (2-tailed)	,116	.
		N	9	9

Tabela 23: Resultado do coeficiente de correlação de Spearman para a medida Precision usando Classes Manuais (CM) e Classes Automáticas (CA).

Correlations

			CM_Precision	CA_Precision
Spearman's rho	CM_Precision	Correlation Coefficient	1,000	,711*
		Sig. (2-tailed)	.	,032
		N	9	9
	CA_Precision	Correlation Coefficient	,711*	1,000
		Sig. (2-tailed)	,032	.
		N	9	9

*. Correlation is significant at the 0.05 level (2-tailed).

Tabela 24: Resultado do coeficiente de correlação de Spearman para a medida Recall usando Classes Manuais (CM) e Classes Automáticas (CA).

Correlations

			CM_Recall	CA_Recall
Spearman's rho	CM_Recall	Correlation Coefficient	1,000	,723*
		Sig. (2-tailed)	.	,028
		N	9	9
	CA_Recall	Correlation Coefficient	,723*	1,000
		Sig. (2-tailed)	,028	.
		N	9	9

*. Correlation is significant at the 0.05 level (2-tailed).

Relativamente à medida *Rand Index*, confirma-se uma forte correlação positiva entre as Classes Manuais (CM) e as Classes automáticas (CA) com rs=0,861 e p=0,003, comprovando que é estatisticamente significante (Tabela 25).

Tabela 25: Resultado do coeficiente de correlação de Spearman para a medida Rand Index usando Classes Manuais (CM) e Classes Automáticas (CA).

Correlations

			CM_RI	CA_RI
Spearman's rho	CM_RI	Correlation Coefficient	1,000	,861**
		Sig. (2-tailed)	.	,003
		N	9	9
	CA_RI	Correlation Coefficient	,861**	1,000
		Sig. (2-tailed)	,003	.
		N	9	9

**. Correlation is significant at the 0.01 level (2-tailed).

Por fim, a medida *Purity* (Tabela 26) apresenta a correlação mais forte entre as classes automáticas e manuais, sendo esta correlação significante pois sendo $\alpha = 0{,}01,\ p < \alpha$.

Tabela 26: Resultado do coeficiente de correlação de Spearman para a medida Purity usando Classes Manuais (CM) e Classes Automáticas (CA).

Correlations

			CM_Purity	CA_Purity
Spearman's rho	CM_Purity	Correlation Coefficient	1,000	,962**
		Sig. (2-tailed)	.	,000
		N	9	9
	CA_Purity	Correlation Coefficient	,962**	1,000
		Sig. (2-tailed)	,000	.
		N	9	9

**. Correlation is significant at the 0.01 level (2-tailed).

4.5.4. Comparando os Resultados do Algoritmo *k-means++* com o *k-means++* com *Tags*

Observando a Tabela 27 e a Tabela 28 podemos ver que, quando usamos apenas os resumos dos artigos, os resultados das medidas de avaliação externa são piores em comparação com os resultados obtidos quando se utiliza o texto total dos artigos. Assim, menos informação parece originar *clusters* piores.

Na Tabela 27 em D1 os melhores resultados são obtidos para o algoritmo *k-means*++ sem integração das *tags*. Apesar disso, na Tabela 28, no repositório DA_1 verifica-se uma melhoria moderada quando o parâmetro *Social Slider* (SS) é igual a 5 e 30. De facto, a medida *Purity* obtém os melhores resultados com SS=30 e portanto, significa que 71% dos documentos estão organizados como as classes manuais (CM) e 67% como nas classes automáticas (CA).

Tabela 27: Resultados da avaliação do algoritmo k-means++ com e sem integração de *tags*, usando classes manuais (CM) e classes automáticas (CA) para os repositórios D_1, D_2 e D_3.

	SS	**Classes**	**F_1**	***Precision***	***Recall***	**RI**	***Purity***
D_1	**0**	CA	0,70	0,60	0,85	0,87	0,92
		CM	0,60	0,53	0,76	0,85	0,85
	5	CA	0,40	0,30	0,56	0,81	0,65
		CM	0,50	0,40	0,57	0,82	0,60
	30	CA	0,41	0,34	0,51	0,78	0,78
		CM	0,54	0,46	0,64	0,82	0,75
D_2	**0**	CA	0,57	0,55	0,59	0,82	0,78
		CM	0,49	0,43	0,57	0,80	0,73
	5	CA	0,82	0,78	0,86	0,92	0,88
		CM	0,63	0,54	0,74	0,85	0,84
	30	CA	0,55	0,40	0,85	0,73	0,88
		CM	0,45	0,32	0,77	0,69	0,84
D_3	**0**	CA	0,50	0,44	0,57	0,79	0,75
		CM	0,50	0,43	0,60	0,80	0,75
	5	CA	0,43	0,36	0,55	0,74	0,68
		CM	0,42	0,34	0,57	0,74	0,70
	30	CA	0,78	0,68	0,91	0,91	0,93
		CM	0,71	0,60	0,87	0,88	0,90

No D_2, Tabela 27, os melhores resultados são obtidos quando SS=5, e em particular quando os *clusters* são comparados com as classes automáticas (CA), verificando-se que 88% dos documentos estão organizados da mesma forma que nas classes automáticas (CA) (*Purity*). Verifica-se ainda que a percentagem de pares de documentos corretamente associados é de 92% (*Rand Index*). F1 é 80% o que evidencia que a média harmónica entre a percentagem de pares de documentos que estão corretamente associados ao mesmo *cluster* (*Precision*) e a percentagem de pares que estão corretamente associados ao mesmo *cluster* de entre os pares que estão ou deveriam estar no mesmo *cluster*

(*Recall*). Por outro lado, na Tabela 28, não vemos nenhuma melhoria significativa com a introdução das *tags*.

Finalmente, quando analisamos o repositório D_3 (Tabela 27) os melhores resultados são obtidos quando SS=30, com resultados similares aos obtidos em D_2 para SS=5. Na Tabela 28, para o repositório DA_3 podemos verificar que os melhores resultados também são obtidos quando SS=30 com F1=60%; *Rand Index*= 82% e *Purity*=81% quando os *clusters* são comparados com as classes automáticas (AC) e F1=59%; *Rand Index*=82% e *Purity*=86% quando os *clusters* são comparados com as classes manuais (CM).

A utilização do *k-means*++ com integração de *tags* por vezes providencia melhores resultados quando comparado com o *k-means*++ sem integração de *tags* ainda que esta melhoria não seja diretamente proporcional à escolha do parâmetro SS – por vezes SS=5 produz piores resultados do que quando não há integração de *tags*; outras, produz melhores resultados do que quando é utilizado SS=30.

Tabela 28: Resultados da avaliação do algoritmo k-means++ com e sem integração de tags, usando classes manuais (MC) e classes automáticas (CA) para os datasets DA1, DA2 e DA3.

	SS	**Classes**	**F_1**	***Precision***	***Recall***	**RI**	***Purity***
DA_1	**0**	CA	0,34	0,31	0,37	0,74	0,63
		CM	0,35	0,31	0,39	0,75	0,56
	5	CA	0,41	0,36	0,48	0,71	0,60
		CM	0,36	0,29	0,49	0,72	0,65
	30	CA	0,44	0,42	0,46	0,77	0,67
		CM	0,45	0,40	0,51	0,79	0,71
DA_2	**0**	CA	0,50	0,47	0,53	0,78	0,76
		CM	0,48	0,41	0,58	0,79	0,71
	5	CA	0,52	0,54	0,51	0,79	0,69
		CM	0,54	0,48	0,62	0,83	0,73
	30	CA	0,55	0,53	0,57	0,79	0,73
		CM	0,46	0,39	0,57	0,78	0,71
DA_3	**0**	CA	0,41	0,33	0,56	0,74	0,71
		CM	0,31	0,31	0,55	0,74	0,71
	5	CA	0,32	0,27	0,38	0,73	0,57
		CM	0,29	0,24	0,36	0,72	0,55
	30	CA	0,60	0,48	0,79	0,82	0,81
		CM	0,59	0,47	0,82	0,82	0,86

4.5.5. Análise Comparativa do Algoritmo *k-means++* com o Algoritmo *k-Communities* (K-C)

Nesta secção procedemos a uma análise comparativa dos resultados obtidos pelo algoritmo k-means++ e o algoritmo k-C, proposto na secção 3.3.

Na Tabela 29, apresentamos os resultados das medidas de avaliação externa, F1, *Precision*, *Recall*, *Rand Index* e *Purity* para os repositórios que utilizam todo o texto e na Tabela 30 sintetizamos a informação recolhida apresentando a média para cada uma das medidas.

Tabela 29: Resultados das medidas de avaliação externa para os algoritmos k-means++ e k-C, usando os repositórios D_1, D_2 e D_3.

Repositório	Algoritmo de *Clustering*	Classes	F_1	Precision	Recall	RI	Purity
D_1	***k-means++***	CA	0,7	0,6	**0,85**	0,87	**0,92**
		CM	0,6	0,53	**0,76**	0,85	**0,85**
	k-C	CA	0,7	**0,84**	0,6	**0,92**	0,79
		CM	**0,7**	**0,88**	0,58	**0,92**	0,71
D_2	***k-means++***	CA	0,57	0,55	0,59	0,82	0,78
		CM	0,49	0,43	0,57	0,8	0,73
	k-C	CA	**0,82**	**0,97**	**0,71**	**0,93**	**0,86**
		CM	**0,74**	**0,79**	**0,69**	**0,92**	**0,82**
D_3	***k-means++***	CA	0,5	0,44	0,57	0,79	0,75
		CM	0,5	0,43	0,6	0,8	0,75
	k-C	CA	**0,81**	**0,83**	**0,79**	**0,94**	**0,91**
		CM	**0,95**	**0,97**	**0,93**	**0,98**	**0,97**

Assim, em média os resultados da implementação do algoritmo k-C são melhores do que os resultados obtidos pelo algoritmo *k-means++*. Isto acontece quer na comparação dos resultados de *Clustering* com as classes manuais (CM), quer com as classes automáticas (CA) (criadas através do algoritmo da "*Ground Truth* Automática").

A média do *Recall* indica que o algoritmo k-C tem um menor número de Falsas Negativas, dito de outro modo, há em média um menor número de pares que pertencem a *clusters* diferentes e que deviam fazer parte do mesmo *cluster*.

Olhando agora para o conteúdo de cada *cluster* podemos ver que o algoritmo k-C providencia os melhores resultados para a média dos valores da medida *Precision*, isto é, há mais pares de documentos que estão corretamente associados no mesmo *cluster*. Observando-se uma melhoria de cerca de 35%, comparando com o algoritmo *k-means*++.

É pertinente também observar que em média houve uma aumento de aproximadamente 10% de decisões corretas (*Rand Index*), isto é, Verdadeiras Positivas e Verdadeiras Negativas, quando se utiliza o algoritmo k-C.

Por fim, analisando os resultados da medida *Purity*, observamos que existe um aumento quando utilizamos o algoritmo k-C, ainda que menos significativo (em média 2% quando são utilizadas as classes automáticas e de 6% quando são utilizadas as classes manuais).

Tabela 30: Média dos Resultados das medidas de avaliação externa para os algoritmos k-means++ e k-C, usando os repositórios D1, D2 e D3.

Medidas de Avaliação Externa	**Algoritmos de *Clustering***			
	***k-means*++**		**k-C**	
	CA	**CM**	**CA**	**CM**
F_1	0.59	0.53	**0.78**	**0.80**
Precision	0.53	0.46	**0.88**	**0.88**
Recall	0.67	0.64	**0.70**	**0.73**
Rand Index	0.83	0.82	**0.93**	**0.94**
Purity	0.82	0.78	**0.85**	**0.83**

Analisando agora os resultados dos repositórios que utilizam apenas os resumos dos artigos, Tabela 31 (resultados completos) e Tabela 32 (média dos resultados), podemos constatar que foi o algoritmo k-C que obteve sempre os melhores resultados.

Como já tinha sido referido anteriormente, os resultados do algoritmo *k-means*++ quando utiliza todo o texto forma *clusters* mais próximos das classes manuais e automáticas do que quando utiliza apenas os resumos dos artigos. Contudo, o algoritmo k-C mostra resultados mais consistentes, não se observando diferenças significativas entre a utilização de todo o texto e dos resumos, indicando que este algoritmo é mais estável.

Tabela 31: Resultados das medidas de avaliação externa para os algoritmos k-means++ e k-C, usando os repositórios DA1, DA2 e DA3.

Repositório	Algoritmo de *Clustering*	Classes	F_1	Precision	Recall	RI	Purity
DA_1	k-means++	CA	0.34	0.31	0.37	0.74	0.63
		CM	0.35	0.31	0.39	0.75	0.56
	k-C	CA	**0.68**	**0.87**	**0.56**	**0.91**	**0.75**
		CM	**0.56**	**0.58**	**0.55**	**0.85**	**0.73**
DA_2	k-means++	CA	0.50	0.47	0.53	0.78	0.76
		CM	0.48	0.41	0.58	0.79	0.71
	k-C	CA	**0.75**	**0.89**	**0.65**	**0.91**	**0.80**
		CM	**0.76**	**0.86**	**0.69**	**0.91**	**0.86**
DA_3	K-means++	CA	0.41	0.33	0.56	0.74	0.71
		CM	0.31	0.31	0.55	0.74	0.71
	K-C	CA	**0.83**	**0.78**	**0.89**	**0.94**	**0.88**
		CM	**0.77**	**0.71**	**0.85**	**0.92**	**0.86**

Tabela 32: Média dos Resultados das medidas de avaliação externa para os algoritmos k-means++ e k-C, usando os repositórios DA_1, DA_2 e DA_3.

Medidas de Avaliação Externa	Algoritmos de *Clustering*			
	k-means++		k-C	
	AC	MC	AC	MC
F_1	0.42	0.38	**0.75**	**0.70**
Precision	0.37	0.34	**0.85**	**0.72**
Recall	0.49	0.51	**0.70**	**0.70**
Rand Index	0.75	0.76	**0.92**	**0.89**
Purity	0.70	0.66	**0.81**	**0.82**

4.5.6. Análise Comparativa do Algoritmo *k-means++* com o Algoritmo *k-Communities* (K-C) Com e Sem Integração de *Tags*.

Resta agora analisar se a integração das *tags* no algoritmo k-C tem algum impacto na formação dos *clusters*, para fazer essa comparação vamos utilizar apenas os repositórios com os texto completos dos artigos (D_1, D_2 e D_3) e vamos utilizar apenas as classes manuais para obter os resultados das medidas de avaliação externa.

Na Figura 80, podemos ver que os resultados do algoritmo k-C variam entre 0,5 e 1, enquanto no algoritmo *k-means++* (Figura 81) variam entre 0,3 e 0,9. Isto significa que existe uma maior dispersão de resultados no algoritmo *k-means++.*

No algoritmo *k-means++* podemos ver que existe um maior impacto na integração das *tags*, mais especificamente nos repositórios D_2 e D_3 onde os parâmetros SS=5 e SS=30, respetivamente, providenciam melhores resultados do que quando não existe integração das *tags*.

No algoritmo k-C apenas encontramos melhores resultados para o repositório D_1 quando se utiliza o parâmetro SS=5.

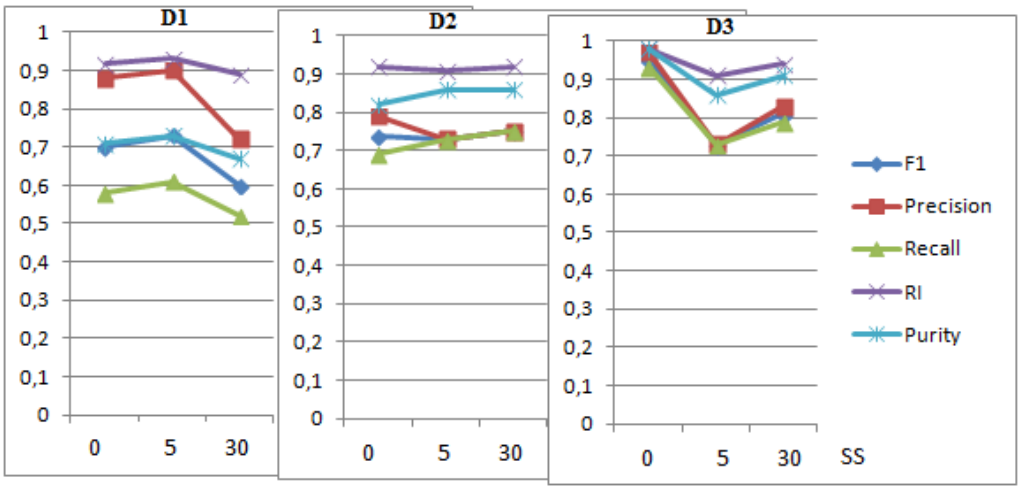


Figura 80: Resultados das medidas de avaliação externa para os repositórios D_1, D_2 e D_3, usando o algoritmo k-C com e sem integração de *tags* (Cunha & Figueira, 2012).

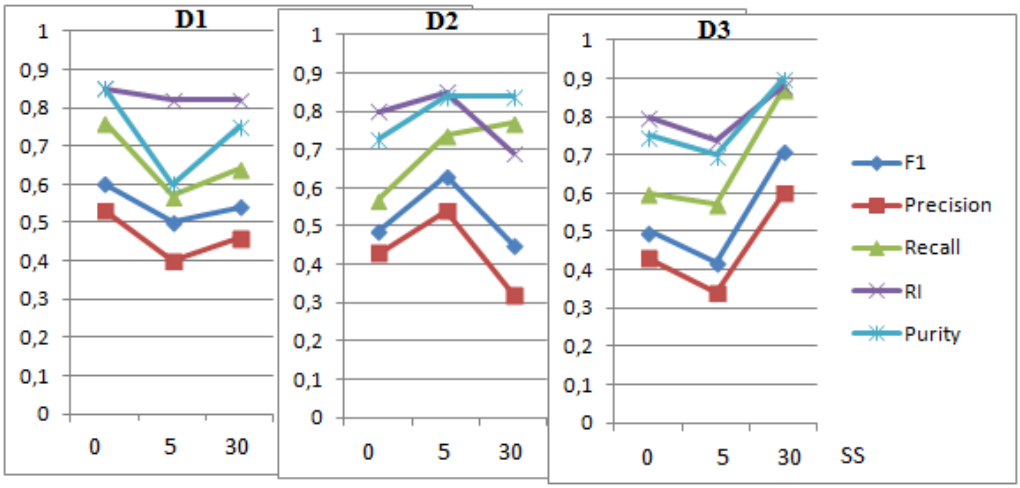


Figura 81: Resultados das medidas de avaliação externa para os repositórios D_1, D_2 e D_3, usando o algoritmo *k-means++* com e sem integração de *tags* (Cunha & Figueira, 2012).

Podemos concluir que a integração das *tags* no algoritmo k-C tem pouco impacto, ainda assim, é este o algoritmo que em média apresenta melhores resultados.

4.5.7. Avaliação Interna

Resta analisar os resultados da medida de avaliação interna MCI, apresentada na secção 4.3.1.

Como se pode ver na Tabela 33, o algoritmo k-C obtém melhores resultados que o algoritmo *k-means*++ e quase todos os testes. O único caso onde o algoritmo *k-means*++ obtém melhor performance é no repositório D2 com SS=5, ainda que a diferença não seja particularmente significante.

Podemos ainda ver que à medida que o parâmetro SS aumenta também aumenta a média da distância ao *cluster* mais próximo, em comparação com as distâncias observadas ao documento mais próximo dentro de cada *cluster*. Isto confirma que utilizar a similaridade dos cossenos e a integração das *tags* para aproximar os documentos que partilham as mesmas *tags* e separa os documentos que não têm *tags* em comum.

Contudo, comparando estes resultados com os resultados das medidas de avaliação externa podemos concluir que apesar do *Maximum Cosine Index* (MCI) indicar uma melhoria quando o parâmetro SS é aumentado, não significa que existe uma melhoria correspondente na qualidade dos *clusters* formados.

Tabela 33: Resultados do Índice MCI

	SS	k-C	k-means++
D_1	0	13,746	3,731
	5	74,923	8,589
	30	400,047	80,630
D_2	0	4,359	1,701
	5	13,621	14,521
	30	181,874	101,966
D_3	0	19,673	5,189
	5	157,450	3,993
	30	1354,626	148,494

4.6. Caso de Estudo II – Repositório de Notícias I – interpretante é o autor das tags

Recentemente Cravino *et al.* (2012) sugeriram uma medida de proximidade pesada baseada na similaridade dos cossenos e que tem em consideração uma rede de *tags*.

O utilizador pode escolher o grau de influência do aspeto social no *Clustering* dos documentos, usando o parâmetro *Social Slider*, que permite dar pesos às *tags*. Adicionalmente, *tags* relacionadas são identificadas através das sobreposições observadas nas comunidades obtidas numa rede de *tags*. Cada vetor de documento é construído e o algoritmo de *clustering k-means* é executado utilizando a similaridade dos cossenos pesada. Os resultados experimentais obtidos pelos autores não identificaram melhorias significativas.

Usando o mesmo repositório pretendemos avaliar se o algoritmo k-C apresenta melhores resultados em comparação com este método.

Esta é uma análise feita na perspetiva do autor das *tags*, uma vez que todos os fragmentos de notícia colecionados por apenas um utilizador.

4.6.1. Descrição do Repositório

O repositório (D_{Clips}) tem 124 clips de notícias colecionados por um utilizador no âmbito do projeto internacional de investigação chamado *Breadcrumbs*[8] (ref. UTA-Est/MAI/0007/2009), onde cada um pode agregar fragmentos de notícias online e associar tags. O utilizador identificou 6 classes: *libya*; *US Tax*; *World Debt Crises*, *Italy downgrading*; *Greece* e outros.

4.6.2. Comparação com o Algoritmo k-C

Na Tabela 34, podemos ver que o algoritmo k-C apresenta os melhores resultados em comparação com o *k-means* para texto e *texto+tags* (usando a similaridade de cossenos pesada).

Como podemos ver na Figura 82 as comunidades obtidas mostram que o utilizador não viu relações entre clips que estão em diferentes comunidades. Contudo, como se mostra na Figura 83, quando se faz a fusão do grafo das *tags* com o grafos das distâncias, onde cada clip de notícia está ligado ao seu clip mais próximo, percebemos que o clip mais próximo de cada documento nem sempre está na mesma comunidade. Mais ainda se salienta que dada a espessura das arestas que ligam estes clips que estão em diferentes

[8] http://breadcrumbs.up.pt/

comunidades indicam que existem clips muito similares e que estão colocados em diferentes comunidades. Contudo é importante salientar que o número de clips que estão nestas circunstancias é muito reduzido, justificando por isso os resultados muito similares obtidos quando são usadas as classes manuais e as classes automáticas.

Tabela 34: Medidas de avaliação externa para o repositório D_{Clips}, usando classes automáticas (CA) e classes manuais (CM)

Medidas de avaliação externa	**Algoritmo de *Clustering***			
	k-means		***k-Communities* (k-C)**	
	Texto	**Texto + Tags**	**CA**	**CM**
F_1	0,27	0,26	0,47	0,49
Precision	0,23	0,24	0,54	0,56
Recall	0,32	0,30	0,42	0,44
Rand Index	0,66	0,67	0,81	0,82

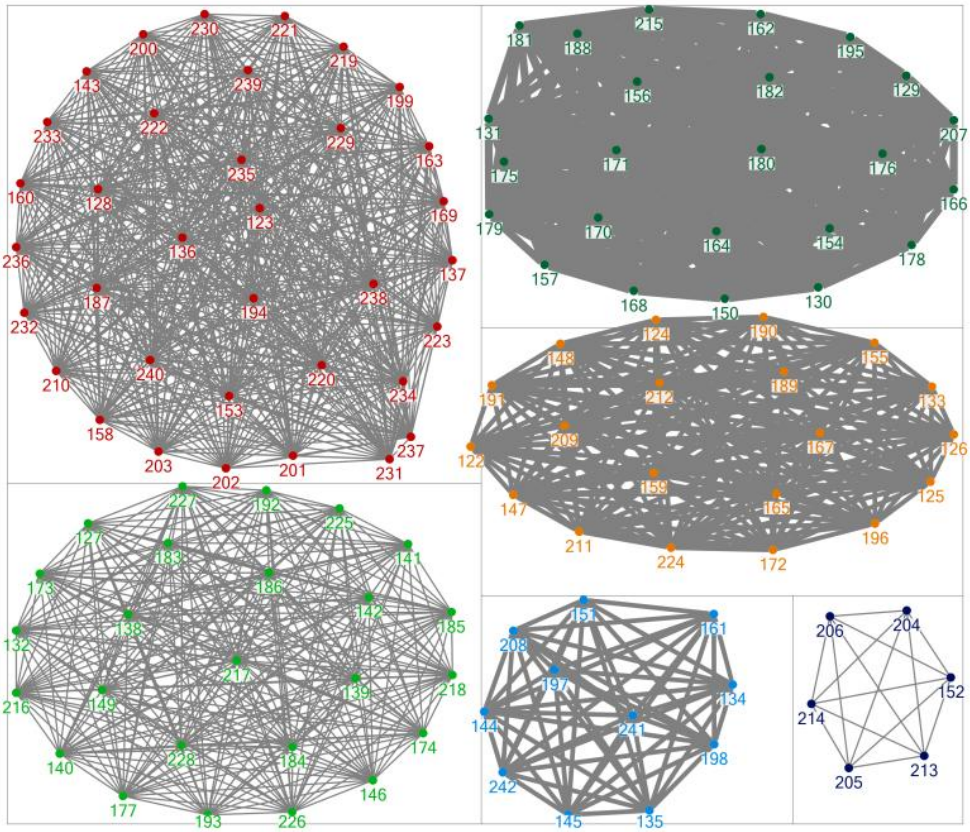


Figura 82: Comunidade obtidas no grafo das *tags* usando o algoritmo Girvan e Newman (Cunha & Figueira, 2012).

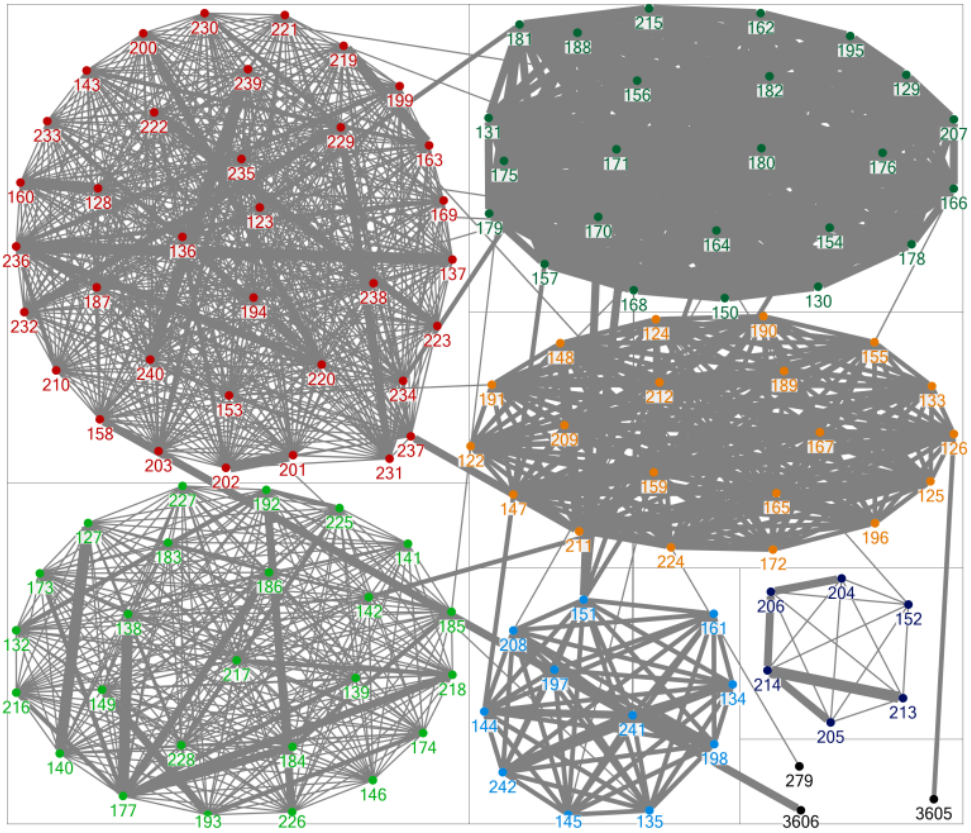


Figura 83: Fusão do grafo das *tags* e do grafo das distâncias (de cada clip ao clip mais próximo) (Cunha & Figueira, 2012).

4.7. Caso de Estudo III – Repositório notícias II– interpretante é o autor das tags

Neste repositório vamos analisar o resultado do *clustering* na perspetiva do autor das *tags* (interpretante). Para isso vamos utilizar o algoritmo k-C, o algoritmo *Spherical k-means*.

4.7.1. Descrição do Repositório

Um utilizador do sistema *Breadcrumbs*, colecionou 65 notícias e associou *tags*, cuja *tag cloud* se apresenta na Figura 84. Do agrupamento manual constam 13 classes, sendo que 4 das classes têm apenas 1 documento, como se pode observar na Tabela 35.

Tabela 35: Classificação manual: número de notícias em cada *cluster*.

Cluster	1	2	3	4	5	6	7	8	9	10	11	12	13
N.º de notícias	13	9	4	3	7	14	5	2	1	1	1	1	4

Figura 84: *Tag cloud* do repositório de notícias

4.7.2. Comparação dos Algoritmos de *Clustering* k-C e *Spherical k-means*

O algoritmo k-C foi executado e em primeiro lugar procedeu-se à seleção das sementes, cujo número é igual ao número de comunidades detetadas com mais de s notícias. Para isso foi utilizado o algoritmo de deteção de comunidades Wakita-Tsurumi (Wakita & Tsurumi, 2007) (o algoritmo Girvan e Newman (Girvan & Newman, 2002) produz as mesmas comunidades) e pela observação da Figura 85, onde os nós representam notícias e as arestas indicam as ligações existentes por terem *tags* que coocorrem nas duas notícias. Das 65 notícias 55 aparecem relacionadas com outras por terem *tags* comuns.

As ligações intracomunidades são quase inexistentes (apenas as comunidades com as *tags Radioactive* e Portugal partilham *tags*). Para além disso, a comunidade mais pequena tem três notícias, não existindo razões para não utilizarmos todas as comunidades apresentadas.

O algoritmo inicia com 7 sementes, as notícias que têm maior grau dentro da sua comunidade, e de seguida são definidas as sementes finais, dependendo da margem mediante a qual devem ser acrescentadas novas sementes. Por exemplo, se escolhermos 0, significa que apenas quando a similaridade dos cossenos entre o vetor que representa a notícia e os vetores de todas as sementes for zero é que esta notícia deve ser acrescentada à lista das sementes. Para tomarmos esta decisão vamos utilizar como indicador o diagrama de extremos e quartis da similaridade dos cossenos entre cada notícia e a sua notícia mais próxima (Gráfico 2).

Verifica-se que entre os 25% de notícias que estão mais distantes do seu documento mais próximo, essa distância varia entre 0,026 e 0,11.

Como ainda existem 11 notícias que não foram incluídas na deteção de comunidades, 7 *clusters* poderá ser insuficiente. Por isso testamos o algoritmo k-C alterando o parâmetro que define a similaridade dos cossenos mínima a que deve a notícia estar da semente para que possa fazer parte desse *cluster*.

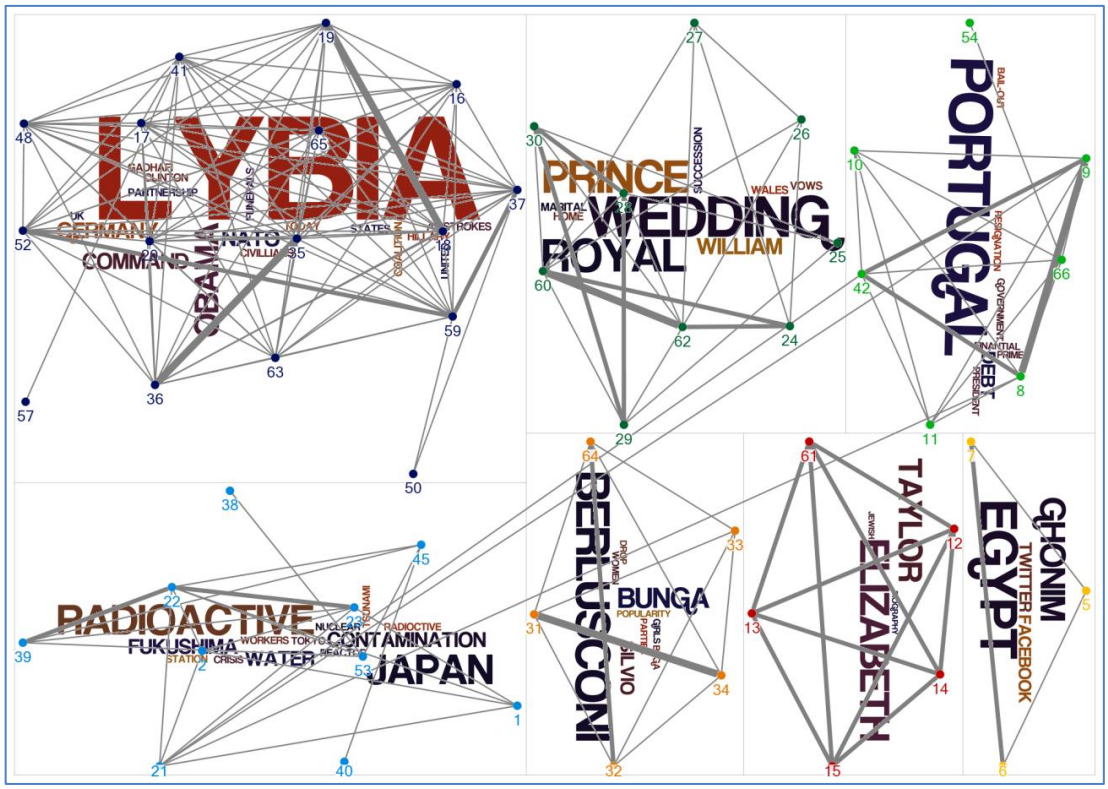


Figura 85: Deteção de comunidades através do algoritmo Wakita-Tsurumi obtido através do software NodeXL, juntamente com as *tags* correspondentes a cada comunidade.

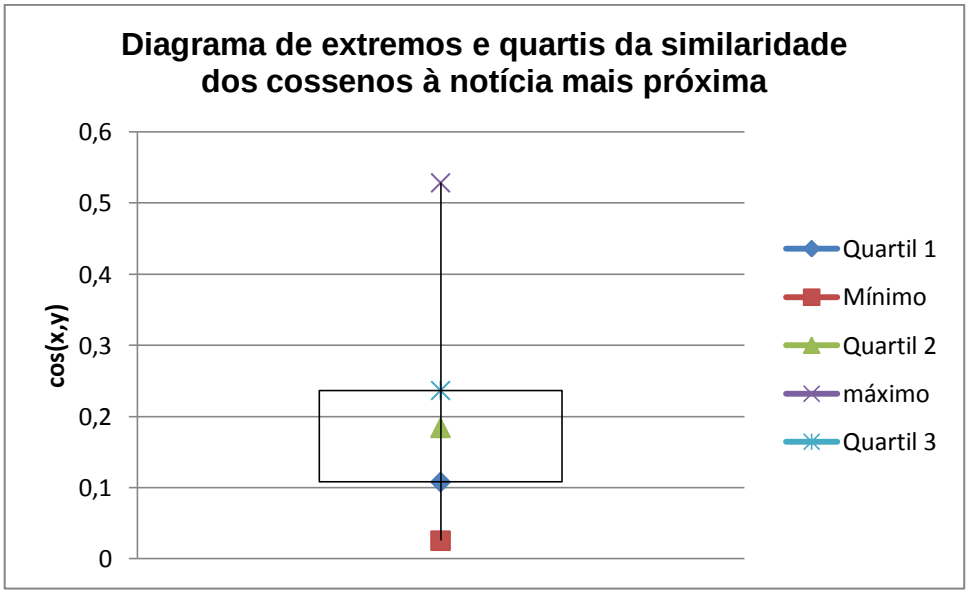


Gráfico 2: Gráfico de extremos e quartis da similaridade dos cossenos à notícia mais próxima.

Observando a Tabela 36, verifica-se que se a margem para acrescentar novas sementes for 0, não existe qualquer alteração ao número de *clusters*. Portanto, com base na informação obtida através do diagrama de extremos e quartis a próxima margem escolhida foi de 0,03, o que nos permitiu obter 9 *clusters*. Quando utilizámos a margem 0,05 ficámos com 12 *clusters* mas quando alteramos o parâmetro para 0,06 voltamos a obter 10 *clusters*.

Tabela 36: Resultados das medidas de avaliação externa do repositório Notícias, usando o algoritmo k-C

	Algoritmo k-C					
Margem	**0**	**0,03**	**0,035**	**0,04**	**0,05**	**0,06**
k	**7**	**9**	**10**	**11**	**12**	**10**
F1	0,571429	0,575289575	**0,640657**	0,590517	0,608501	0,554622
Precision	0,501475	0,56870229	0,675325	0,658654	**0,712042**	0,6
Recall	0,664063	0,58203125	**0,609375**	0,535156	0,53125	0,515625
Rand Index	0,881119	0,897435897	**0,918415**	0,911422	**0,918415**	0,901166
Purity	**0,818182**	0,787878788	0,772727	0,727273	0,727273	0,727273

Portanto, em termos de número de *clusters*, o que apresenta 12 *clusters* é o que mais se aproxima do número de agrupamento escolhidos na organização manual (13). Contudo, é o *clustering* com 10 *clusters* e margem 0,035 que apresenta melhores resultados para as medidas F1, *Recall* e *Rand Index*. Em relação à medida *Purity*, esta apresenta os melhores resultados quando são utilizados 7 *clusters*, e a medida *Precision* quando são utilizados 12 *clusters*. Com base nesta informação optamos por executar o algoritmo *Spherical k-means* com 5 runs para k=10 e k=12 e k=13.

Tabela 37: Resultados das medidas de avaliação externa do repositório Notícias, usando o algoritmo *Spherical k-means* com k=13.

k=13	**run 1**	**run 2**	**run 3**	**run 4**	**run 5**	**Média**
F1	0,591017	0,392344	0,326721	0,417978	0,335221	0,412656
Precion	0,702247	0,460674	0,380435	0,492063	0,377551	0,482594
Recall	0,510204	0,341667	0,286299	0,363281	0,301426	0,360575
RI	0,919347	0,881585	0,865501	0,879254	0,86317	0,881772
Purity	0,712121	0,590909	0,545455	0,590909	0,560606	0,6

Na Tabela 37, podemos ver os resultados para a execução do algoritmo *Spherical k-means* com 13 *clusters*. Apenas no *run* 1 os resultados se aproximam dos obtidos pelo

algoritmo k-C. Para os restantes *runs* os resultados são significativamente piores. A média dos resultados de cada medida de avaliação externa é inferior a qualquer dos resultados obtidos nos vários testes executados no algoritmo k-C com diferentes iniciações do algoritmo. Apenas na perspetiva da medida *Rand Index* não se verificam diferenças significativas.

De seguida, testamos o algoritmo *Spherical k-means* para k=12, igualmente com 5 runs, e neste caso podemos comparar diretamente com k=12 do algoritmo k-C, uma vez que este k foi testado (Tabela 38). Constatamos que os resultados médios de cada uma das medidas de avaliação do algoritmo *Spherical k-means* são significativamente inferiores aos resultados obtidos pelo algoritmo k-C.

Tabela 38: Resultados das medidas de avaliação externa do repositório Notícias, usando o algoritmo *Spherical k-means* com k=12.

K=12	***run* 1**	***run* 2**	***run* 3**	***run* 4**	***run* 5**	**Média**
F1	0,518359	0,399494	0,437718	0,366337	0,323926	0,409167
Precision	0,55814	0,461988	0,489583	0,45122	0,381503	0,468487
Recall	0,483871	0,351893	0,395789	0,308333	0,28145	0,364267
RI	0,896037	0,889277	0,887413	0,880653	0,871562	0,884988
Purity	0,681818	0,560606	0,515152	0,5	0,545455	0,560606

Por último, executamos o algoritmo *Spherical k-means* com k=10 porque foi no algoritmo k-C, o que obteve melhores resultados.

Tabela 39: Resultados das medidas de avaliação externa do repositório Notícias, usando o algoritmo *Spherical k-means* com k=10.

k=10	***run* 1**	***run* 2**	***run* 3**	***run* 4**	***run* 5**	**Média**
F1	0,511797	0,331313	0,259124	0,312	0,356713	0,354189
Precion	0,477966	0,343096	0,243151	0,319672	0,366255	0,350028
Recall	0,550781	0,320313	0,277344	0,304688	0,347656	0,360156
RI	0,874592	0,845688	0,810723	0,839627	0,85035	0,844196
Purity	0,727273	0,560606	0,560606	0,606061	0,590909	0,609091

Verificamos que os melhores resultados foram obtidos no *run* 1, ainda assim existe uma diferença de aproximadamente 13% em relação à medida F1, 20% na medida *Precision*, 6% na medida *Recall*. Contudo, se compararmos com a média dos resultados obtidos nos 5 runs, estas diferenças são mais significativas, ou seja, 28,6% na medida F1 que é a

média harmónica do *Precision* e do *Recall*. No algoritmo k-C há mais 32,5% de pares que estão corretamente colocados nos seus *clusters* (*Precision*) e 24,9% dos pares que estão corretamente colocados nos *clusters* considerando os pares que estão e os que deveriam estar (*Recall*). Existem ainda mais 7% de decisões corretas (Rand Index). Por fim, há aproximadamente mais 16,4% de documentos que estão organizados como na classificação manual (Purity).

4.8. Caso de Estudo IV – Wikipedia – interpretante é a comunidade de utilizadores

Utilizando este repositório vamos analisar o resultado do *clustering* na perspetiva da comunidade de utilizadores (interpretante). Para isso vamos utilizar o algoritmo k-C e o algoritmo *Spherical k-means*. Os algoritmos *k-means* e *k-means++* não foram considerados para analisar os resultados uma vez que se verificou não funcionarem corretamente para este tipo de dados (geralmente colocam quase todos os documentos num *cluster*, deixando os restantes com 1 documento).

O repositório D_{wiki} é constituído por 1170 documentos e será dividido em 10 repositórios estratificados: um será usada para o teste e os restantes são fundidos e usados para o treino. O procedimento é repetido 10 vezes. A esta técnica estatística chamamos de 10-*fold cross validation* estratificada. Segundo Witten e Frank (2005), extensivos testes em numerosas bases de dados mostram que 10 é o número indicado de partições para obter uma boa estimativa do erro, existindo algumas evidências teóricas que o suportam. Contudo, os autores alertam que a escolha de 5 ou 20 para k podem ser igualmente boas escolhas.

O nosso objetivo é comparar os resultados dos dois algoritmos quer para os dados do teste quer para os dados do treino e no final determinar qual dos algoritmos é mais estável. Para isso vamos recorrer à formulação de testes de hipóteses utilizando como teste estatístico o teste Wilcoxon *Signed Ranks*. A escolha deste teste baseou-se na reflexão feita por Demsar aos testes estatísticos utilizados para comparar classificadores (Demsar, 2006), na medida em que, sendo este um teste não paramétrico, permite uma comparação pareada dos resultados não exigindo que os estes sigam uma distribuição normal e é adequado ao número de resultados que temos disponíveis (só com mais de 30 dados e com a garantia de que a população é normal é que podemos usar o *pared t-test*, um teste mais potente que o teste de Wilcoxon).

4.8.1. Descrição do Repositório

O repositório *Wikipedia* foi obtida online[9], sendo constituída por 20764 documentos aos quais foram atribuídas *tags* em média por 34,5 utilizadores. Deste repositório recolhemos 12000 documentos e geramos a *tag cloud* que apresentamos na Figura 86. A *tag* "*wikipedia"* é aquela que foi atribuída com maior frequência, ainda que neste contexto seja a *tag* menos importante para detetar as comunidades de documentos que partilham as mesmas *tags*.

Figura 86: *Tag cloud* do repositório *Wikipedia* com 12000 *wikis*.

De seguida, optámos por reduzir o repositório recolhendo apenas os documentos que continham entre as 5 *tags* mais vezes atribuídas as *tags*: "*art"*, "*biology"*, "*health"*, "*physics"*, "*programming"* ou "*typography"*. O repositório atual é constituído por 1170 documentos e a sua *Tag Cloud* está apresentada na Figura 87, verificando-se mais uma vez que é a *tag "wikipedia"* que aparece com maior destaque.

Figura 87: *Tag cloud* do repositório da *Wikipedia* com tags *art*, *biology*, *health*, *physics*, *programming* e *typography*.

[9] http://nlp.uned.es/social-tagging/wiki10+/

Como a *tag "wikipedia"* não é relevante para a organização dos documentos, uma vez que neste caso o nosso interesse não é organizar os documentos de acordo com a sua proveniência (por exemplo, se são provenientes do *Delicious*, da *Wikipedia*, do *Flickr*, etc.), optámos por eliminar as *tags* "*wiki*" e "*wikipedia*" do nosso repositório obtendo como resultado a *tag cloud* apresentada na Figura 88. É fácil perceber que existe uma grande percentagem de documentos aos quais foi atribuída a *tag* "*programming*". Por outro lado, de entre as *tags* que selecionámos para recolher os documentos, a *tag* "*typography*" é a que aparece com menor destaque, evidenciando a existência de menos documentos com esta *tag*.

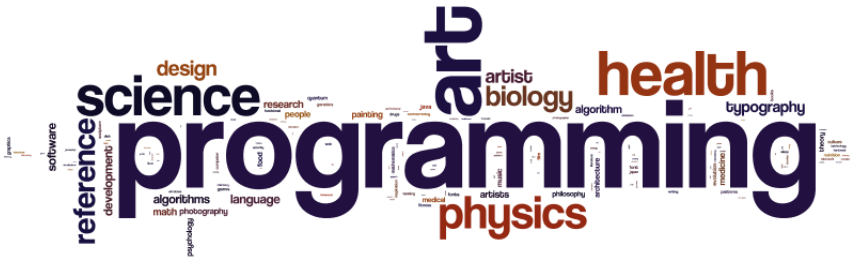


Figura 88: *Tag cloud* do repositório reduzido sem as *tags wiki* e *wikipedia*.

Tabela 40: Classes para o repositório D_{wiki}.

Classes	# Doc	Tag Cloud
Art	280	
Biology	90	
Health	200	
Physics	150	
Programming	400	
Typography	50	

Definido o repositório e as respetivas *tags*, organizámos manualmente os documentos através das *tags* atribuídas pelos utilizadores no sentido de formar 6 classes: "*art*",

"*biology*", "*health*", "*physics*", "*programming*" e "*typography*". Sempre que um documento tinha mais do que uma destas tags atribuída, era colocado na classe cuja *tag* tinha sido mais vezes atribuída pelos utilizadores. Na Tabela 40 podemos ver o número de documentos que ficou em cada classe e a respetiva *tag cloud*.

4.8.2. Resultados dos Dados de Teste

Os 10 repositórios de teste são constituídas por 117 documentos cada um. A Tabela 41 apresenta os resultados das medidas de avaliação externa para o algoritmo *Spherical k-means* e na Tabela 42 podemos observar os resultados do algoritmo k-C usando como critério para acrescentar novos centroides $cos(x_i, C_i) = 0 \; \forall C_i$.

Uma vez que as classes produzidas manualmente são 6, o *Spherical k-means* foi executado com k=6. Relativamente ao algoritmo k-C, o critério escolhido para acrescentar novos centroides não alterou o número de sementes detetadas pelo algoritmo de deteção de comunidades, Wakita-Tsurumi. Assim, o número de sementes variou entre 4 e 5 (em oposição às 6 obtidas manualmente), o que se explica quer pelo facto dos documentos com as *tags* "*biology*" e "*physics*" serem colocados na mesma comunidade (consultado a Tabela 40 verificamos que a *tag* "*science*" aparece com grande destaque nos documentos em que aparecem as *tags* "biology" e "physics", o que levou a que os documentos fossem agrupados numa só comunidade, gerando menos uma classe que as produzidas manualmente) quer por, em 50% dos casos, os documentos com a *tag* "*typography*" aparecem numa só comunidade, sendo que nos restantes casos ficam juntamente com os documentos que têm a *tag* "*art*" (justificando a oscilação entre uma a duas classes a menos que as classes produzidas manualmente).

Analisando os resultados obtidos podemos observar que o algoritmo k-C apresenta em média melhores resultados que o algoritmo *Spherical k-means*. Apenas na medida *Precision* essa diferença é pouco significativa. Relembramos que é natural que isto aconteça, uma vez que o algoritmo k-C forma entre 4 e 5 *clusters* e é comparado com 6 classes, logo inevitavelmente há documentos que ficam no mesmo *cluster* e que na perspetiva da organização manual não deviam estar juntos.

De seguida fazemos a análise da diferença entre os resultados obtidos pelos dois algoritmos para cada uma das medidas de avaliação externa, indicando se são detetadas diferenças significativas entre os algoritmos na perspetiva de cada uma das medidas de avaliação externa. Para isso vamos utilizar o teste de Wilcoxon ($\alpha = 0{,}05$), aplicado a um teste de hipóteses.

Tabela 41: Resultados do algoritmo *Spherical k-means* para os dados de teste.

Repositórios de Teste	***Spherical k-means***				
	F1	***Precision***	***Recall***	***Rand Index***	***Purity***
$\mathbf{D_{1\text{-}teste}}$	0,400379	0,428571	0,375666	0,76344287	0,586538
$\mathbf{D_{2\text{-}teste}}$	0,395519	0,437972	0,360568	0,76829724	0,567308
$\mathbf{D_{3\text{-}teste}}$	0,397032	0,447324	0,356905	0,7605364	0,547009
$\mathbf{D_{4\text{-}teste}}$	0,375462	0,420878	0,338893	0,75095785	0,512821
$\mathbf{D_{5\text{-}teste}}$	0,417417	0,477253	0,370914	0,77129384	0,538462
$\mathbf{D_{6\text{-}teste}}$	0,378947	0,415605	0,348232	0,74786325	0,512821
$\mathbf{D_{7\text{-}teste}}$	0,478657	0,52818	0,437625	0,78941939	0,606838
$\mathbf{D_{8\text{-}teste}}$	0,335244	0,361949	0,312208	0,72649573	0,547009
$\mathbf{D_{9\text{-}teste}}$	0,422272	0,480034	0,376918	0,77217801	0,521368
$\mathbf{D_{10\text{-}teste}}$	0,353976	0,374535	0,335557	0,72944297	0,478632
Média	**0,395491**	**0,43723**	**0,361349**	**0,757993**	**0,54188**

Tabela 42: Resultados do algoritmo k-C para os dados de teste.

Repositórios de Teste	**k-C**				
	F1	***Precision***	***Recall***	***Rand Index***	***Purity***
$\mathbf{D_{1\text{-}teste}}$	0,598482	0,543936	0,665187	0,81235997	0,769231
$\mathbf{D_{2\text{-}teste}}$	0,647297	0,552417	0,781528	0,82094847	0,817308
$\mathbf{D_{3\text{-}teste}}$	0,616832	0,516667	0,765177	0,79000884	0,820513
$\mathbf{D_{4\text{-}teste}}$	0,428205	0,412091	0,44563	0,73710581	0,632479
$\mathbf{D_{5\text{-}teste}}$	0,526753	0,402538	0,761841	0,69761273	0,820513
$\mathbf{D_{6\text{-}teste}}$	0,408938	0,377828	0,44563	0,71544356	0,615385
$\mathbf{D_{7\text{-}teste}}$	0,541971	0,498044	0,594396	0,7780725	0,709402
$\mathbf{D_{8\text{-}teste}}$	0,549521	0,527284	0,573716	0,79221927	0,717949
$\mathbf{D_{9\text{-}teste}}$	0,32967	0,339703	0,320213	0,71234895	0,495726
$\mathbf{D_{10\text{-}teste}}$	0,513191	0,484866	0,54503	0,77158856	0,735043
Média	**0,516086**	**0,465537**	**0,589835**	**0,762771**	**0,713355**

Teste de Hipóteses:

H0: Não há diferenças entre os resultados da medida de avaliação externa quando se utiliza o algoritmo *Spherical k-means* e o algoritmo k-C.

H1: Há diferenças entre os resultados da medida de avaliação externa quando se utiliza o algoritmo *Spherical k-means* e o algoritmo k-C.

a. F1 – Resultados dos dados de teste

Observando a Tabela 43, podemos constatar que em apenas um repositório o resultado da média harmónica do *Precision* e do *Recall* foi melhor para o algoritmo *Spherical k-means*. Como não há empates, o algoritmo k-C apresentou os melhores resultados desta medida para os restantes 9 repositórios.

Tabela 43: Tabela de *Ranks* obtida pelo SPSS para o Teste de Wilcoxon – F1 – dados de teste.

Ranks

		N	Mean Rank	Sum of Ranks
F1kC - F1Skmeans	Negative Ranks	1[a]	4,00	4,00
	Positive Ranks	9[b]	5,67	51,00
	Ties	0[c]		
	Total	10		

a. F1kC < F1Skmeans

b. F1kC > F1Skmeans

c. F1kC = F1Skmeans

Examinada a tabela do teste estatístico, Tabela 44, podemos constatar que a hipótese nula deve ser rejeitada, uma vez que a probabilidade de não haver diferenças entre os dois algoritmos, $p = 0{,}017$, é inferior a α.

Tabela 44: Teste estatístico obtido pelo SPSS para o Teste de Wilcoxon – F1 – dados de teste.

Test Statistics[a]

	F1kC - F1Skmeans
Z	-2,395[b]
Asymp. Sig. (2-tailed)	,017

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

b. *Precision* – Resultados dos dados de teste

Analisando agora os resultados obtidos pela medida *Precision*, Tabela 45, verificamos que em metade dos repositórios foi o algoritmo *Spherical k-means* que apresentou a maior percentagem de documentos corretamente colocados nos *clusters* e na outra metade dos repositórios foi o algoritmo k-C. A este respeito é importante relembrar que o

algoritmo k-C está a utilizar menos *clusters* que os definidos manualmente e por isso é natural que existam vários pares de documentos que estão a ser contabilizados como estando erradamente por estarem colocados no mesmo *cluster*.

Tabela 45: Tabela de *Ranks* obtida pelo SPSS para o Teste de Wilcoxon – *Precision* – dados de teste.

Ranks

		N	Mean Rank	Sum of Ranks
PrecisionkC - PrecisionSkmeans	Negative Ranks	5[a]	4,00	20,00
	Positive Ranks	5[b]	7,00	35,00
	Ties	0[c]		
	Total	10		

a. PrecisionkC < PrecisionSkmeans

b. PrecisionkC > PrecisionSkmeans

c. PrecisionkC = PrecisionSkmeans

Portanto, pela análise do teste estatístico (Tabela 46), confirmamos que não existem diferenças estatisticamente significantes que permitam rejeitar H0, pois $p > \alpha$.

Tabela 46: Teste estatístico obtido pelo SPSS para o Teste de Wilcoxon – *Precision* – dados de teste.

Test Statistics[a]

	PrecisionkC - PrecisionSkmeans
Z	-,764[b]
Asymp. Sig. (2-tailed)	,445

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

c. *Recall* – Resultados dos dados de teste

A medida *Recall* indica-nos a percentagem de pares de documentos que estão corretamente colocados no mesmo *cluster* de entre o total de pares que realmente estão e os que deviam estar. Na Tabela 47 podemos observar que o algoritmo k-C, foi o que obteve os melhores resultados em 9 dos repositórios.

Para além disso, a análise do resultados apresentados na Tabela 48 indicam que os dois algoritmos são significativamente diferentes, sendo o algoritmo k-C o que apresenta os melhores resultados.

Tabela 47: Tabela de *Ranks* obtida pelo SPSS para o Teste de Wilcoxon – *Recall* – dados de teste.

Ranks

		N	Mean Rank	Sum of Ranks
RecallkC - RecallSkmeans	Negative Ranks	1[a]	1,00	1,00
	Positive Ranks	9[b]	6,00	54,00
	Ties	0[c]		
	Total	10		

a. RecallkC < RecallSkmeans

b. RecallkC > RecallSkmeans

c. RecallkC = RecallSkmeans

Tabela 48: Teste estatístico obtido pelo SPSS para o Teste de Wilcoxon – *Recall* – dados de teste.

Test Statistics[a]

	RecallkC - RecallSkmeans
Z	-2,701[b]
Asymp. Sig. (2-tailed)	,007

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

d. *Rand Index* – Resultados dos dados de teste

Analisando a percentagem de decisões corretas (Verdadeiras Positivas e Verdadeiras Negativas), verificamos que em metade dos repositórios foi o algoritmo k-C que obteve os melhores resultados e na outra metade foi o algoritmo *Spherical k-means*.

Tabela 49: Tabela de Ranks obtida pelo SPSS para o Teste de Wilcoxon – *Rand Index* – dados de teste.

Ranks

		N	Mean Rank	Sum of Ranks
RIkC - RISkmeans	Negative Ranks	5[a]	5,00	25,00
	Positive Ranks	5[b]	6,00	30,00
	Ties	0[c]		
	Total	10		

a. RIkC < RISkmeans

b. RIkC > RISkmeans

c. RIkC = RISkmeans

Como seria expectável verificamos que não existem razões para rejeitar a hipótese nula, uma vez que a probabilidade é de 0,799 muito superior a 0,05 (Tabela 50).

Tabela 50: Teste estatístico obtido pelo SPSS para o Teste de Wilcoxon – *Rand Index* – dados de teste.

Test Statistics[a]

	RIkC - RISkmeans
Z	-,255[b]
Asymp. Sig. (2-tailed)	,799

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

e. *Purity* – Resultados dos dados de teste

Por fim, a medida Purity, indica a percentagem de documentos que estão organizados tal como no agrupamento manual é superior no algoritmo k-C para 9 dos repositórios (Tabela 51).

Tabela 51: Tabela de Ranks obtida pelo SPSS para o Teste de Wilcoxon – *Purity*.

Ranks

		N	Mean Rank	Sum of Ranks
PuritykC - PuritySkmeans	Negative Ranks	1[a]	1,00	1,00
	Positive Ranks	9[b]	6,00	54,00
	Ties	0[c]		
	Total	10		

a. PuritykC < PuritySkmeans

b. PuritykC > PuritySkmeans

c. PuritykC = PuritySkmeans

Pela análise da Tabela 52, comprova-se que esta diferença entre os dois algoritmos é estatisticamente significativa uma vez que a probabilidade é 0,007 inferior a 0,05.

Tabela 52: Teste estatístico obtido pelo SPSS para o Teste de Wilcoxon – *Purity*.

Test Statistics[a]

	PuritykC - PuritySkmeans
Z	-2,703[b]
Asymp. Sig. (2-tailed)	,007

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

4.8.3. Resultados dos Dados de Treino

Os 10 repositórios de treino têm cada um 1053 documentos. A Tabela 53 apresenta os resultados da medidas de avaliação externa para o algoritmo *Spherical k-means* e na Tabela 54 apresentamos os resultados do algoritmo k-C (o critério para acrescentar novos centroides é: $cos(x_i, C_i) = 0 \; \forall C_i$).

Tabela 53: Resultados do algoritmo *Spherical k-means* para os dados de treino.

Repositórios de Treino	*Spherical k-means*				
	F1	***Precision***	***Recall***	***Rand Index***	***Purity***
$\mathbf{D_{1\text{-}treino}}$	0,688393	0,749876	0,636228	0,869353901	0,796771
$\mathbf{D_{2\text{-}treino}}$	0,639418	0,72197	0,573807	0,853209552	0,700855
$\mathbf{D_{3\text{-}treino}}$	0,795505	0,817906	0,774299	0,909705747	0,848053
$\mathbf{D_{4\text{-}treino}}$	0,654615	0,717511	0,601858	0,855946616	0,758784
$\mathbf{D_{5\text{-}treino}}$	0,671434	0,73644	0,616974	0,863038431	0,780627
$\mathbf{D_{6\text{-}treino}}$	0,606583	0,682594	0,545805	0,839412289	0,701804
$\mathbf{D_{7\text{-}treino}}$	0,661108	0,731945	0,602773	0,859830143	0,765432
$\mathbf{D_{8\text{-}treino}}$	0,692718	0,751343	0,64258	0,870693546	0,779677
$\mathbf{D_{9\text{-}treino}}$	0,689484	0,755566	0,634031	0,870466059	0,792972
$\mathbf{D_{10\text{-}treino}}$	0,640738	0,718504	0,578161	0,852940539	0,754036
Média	**0,674**	**0,738365**	**0,620652**	**0,864459682**	**0,767901**

Tabela 54: Resultados do algoritmo k-C para os dados de treino.

Repositórios de Treino	k-C				
	F1	***Precision***	***Recall***	***Rand Index***	***Purity***
$\mathbf{D_{1\text{-}treino}}$	0,592705	0,507544	0,712205	0,777981794	0,744539
$\mathbf{D_{2\text{-}treino}}$	0,546148	0,442818	0,71238	0,731448081	0,795821
$\mathbf{D_{3\text{-}treino}}$	0,710563	0,63473	0,806974	0,85088413	0,861349
$\mathbf{D_{4\text{-}treino}}$	0,682162	0,600304	0,789869	0,833049877	0,833808
$\mathbf{D_{5\text{-}treino}}$	0,699661	0,639533	0,77227	0,849614897	0,835708
$\mathbf{D_{6\text{-}treino}}$	0,726439	0,644405	0,832406	0,85779901	0,855651
$\mathbf{D_{7\text{-}treino}}$	0,706838	0,6311	0,803233	0,848872856	0,860399
$\mathbf{D_{8\text{-}treino}}$	0,746549	0,680437	0,82689	0,872650656	0,872745
$\mathbf{D_{9\text{-}treino}}$	0,727635	0,654161	0,819702	0,860810503	0,865147
$\mathbf{D_{10\text{-}treino}}$	0,737831	0,661554	0,83399	0,865569674	0,867047
Média	**0,687653**	**0,609659**	**0,790992**	**0,834868148**	**0,839221**

Tal como procedemos para os dados de teste, o algoritmo *Spherical k-means* foi executado com $k = 6$. Para o algoritmo k-C, o critério escolhido para acrescentar novos centroides não alterou o número de sementes detetadas pelo algoritmo de deteção de comunidades, Wakita-Tsurumi. O número de sementes foi 4 em todos os 10 conjuntos de dados (em oposição às 6 obtidas manualmente).

Analisando os resultados obtidos podemos observar que, em média, o algoritmo k-C apresenta melhores resultados para a medida *Recall*, sendo a diferença de aproximadamente 17%, significando que o resultado do *clustering* do algoritmo K-C tem menos pares FN (False Negatives) do que o algoritmo *Spherical k-means*. Ao contrário, a medida *Precision* apresenta melhores resultados no algoritmo *Spherical k-means* comparativamente com o algoritmo k-C, sendo a diferença de aproximadamente 13%. Recordamos que estes resultados são coerentes com o facto de no algoritmo k-C existirem menos 2 *clusters* do que o esperado pela organização manual, por isso o resultado do *Precision* no algoritmo k-C está a considerar que os pares dos grupos que ficam fundidos são pares FP (*False Positives*). Os resultados das medidas F1 e *Rand Index* não sugerem diferenças significativas entre os dois algoritmos. Por outro lado, a medida *Purity* indica que é o algoritmo k-C que tem mais documentos organizados da mesma forma que a organização manual, sendo essa diferença de aproximadamente 7%. No sentido de averiguar se as diferenças detetadas são significativas do ponto de vista estatístico procedemos à aplicação do Teste de Wilcoxon para cada uma das medidas de avaliação apresentadas, usando tal como na análise dos dados do teste, o mesmo teste de hipóteses.

a. F1 – Resultados dos dados de treino

Na Tabela 55 verifica-se que o algoritmo *Spherical k-means* é o que obtém melhores resultados em 7 dos repositórios. Relembramos contudo que na medida F1 é dado o mesmo peso ao *Precision* e ao *Recall* e, neste caso, como sabemos que o algoritmo k-C é executado com menos dois *clusters* do que na organização manual, prevê-se que este seja menor do que o obtido se fossem utilizados 6 *clusters*.

Tabela 55: Tabela de *Ranks* obtida pelo SPSS para o Teste de Wilcoxon – F1- dados de treino.

Ranks

		N	Mean Rank	Sum of Ranks
F1Skmeans - F1KC	Negative Ranks	7[a]	4,86	34,00
	Positive Ranks	3[b]	7,00	21,00
	Ties	0[c]		
	Total	10		

a. F1Skmeans < F1KC

b. F1Skmeans > F1KC

c. F1Skmeans = F1KC

Apesar das diferenças encontradas, segundo os resultados obtidos na Tabela 56, não há razões para eliminar a hipótese nula, ou seja, não há razões que fundamentem

diferenças significativas entre os algoritmos tendo em conta esta medida uma vez que $p = 0{,}508 > 0{,}05$.

Tabela 56: Teste estatístico obtido pelo SPSS para o Teste de Wilcoxon – F1 – dados de treino.

Test Statistics[a]

	F1Skmeans - F1KC
Z	-,663[b]
Asymp. Sig. (2-tailed)	,508

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

b. Precision – Resultados dos dados de treino

De acordo com a análise prévia feita no início desta secção, comprova-se que é o algoritmo *Spherical k-means* que obtém os melhores resultados para esta medida e para todos os repositórios utilizados (Tabela 57). Esta diferença é estatisticamente significativa pois p=0,005<0,05 (Tabela 58) conduzindo portanto à eliminação da hipótese nula e aceitação da hipótese alternativa.

Tabela 57: Tabela de *Ranks* obtida pelo SPSS para o Teste de Wilcoxon – *Precision* - dados de treino.

Ranks

		N	Mean Rank	Sum of Ranks
PrecisionSkmeans - PrecisionKC	Negative Ranks	0[a]	,00	,00
	Positive Ranks	10[b]	5,50	55,00
	Ties	0[c]		
	Total	10		

a. PrecisionSkmeans < PrecisionKC

b. PrecisionSkmeans > PrecisionKC

c. PrecisionSkmeans = PrecisionKC

Tabela 58: Teste estatístico obtido pelo SPSS para o Teste de Wilcoxon – *Precision* – dados de treino.

Test Statistics[a]

	PrecisionSkmeans - PrecisionKC
Z	-2,803[b]
Asymp. Sig. (2-tailed)	,005

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

c. *Recall* – Resultados dos dados de treino

Ao contrário do que aconteceu com os resultados da medida *Precision*, os resultados da medida *Recall* indicam que é o algoritmo k-C que obtém os melhores resultados para todos os repositórios (Tabela 59). As diferenças encontradas entre os dois algoritmos são estatisticamente significativas, pois de acordo com a Tabela 60, a probabilidade dos dois algoritmos apresentarem resultados similares é 0,005, inferior a $\alpha = 0{,}05$ e por isso devemos rejeitar H0 e aceitar H1.

Tabela 59: Tabela de *Ranks* obtida pelo SPSS para o Teste de Wilcoxon – *Recall* - dados de treino.

Ranks

		N	Mean Rank	Sum of Ranks
RecallSkmeans - RecallKC	Negative Ranks	10[a]	5,50	55,00
	Positive Ranks	0[b]	,00	,00
	Ties	0[c]		
	Total	10		

a. RecallSkmeans < RecallKC

b. RecallSkmeans > RecallKC

c. RecallSkmeans = RecallKC

Tabela 60: Teste estatístico obtido pelo SPSS para o Teste de Wilcoxon – *Recall* – dados de treino.

Test Statistics[a]

	RecallSkmeans - RecallKC
Z	-2,803[b]
Asymp. Sig. (2-tailed)	,005

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

d. Rand Index – Resultados dos dados de treino

Relativamente aos resultados da medida *Rand Index*, observa-se através da Tabela 61, que é o algoritmo *Spherical k-means* que apresenta os melhores resultados para 7 dos repositórios. Contudo, esta diferença não é estatisticamente significativa pois tal como pode observar na Tabela 62, 0,093>0,05, e portanto não há razões para rejeitar a hipótese nula.

É ainda necessário considerar que, tal como anteriormente referido, quando se comparam duas estruturas (a manual e automática) em que a automática gera menos grupos que a manual, há necessariamente um conjunto de pares que em vez de serem

TP (*True Positives*) são FP (*False Positives*). Este facto tem impacto no cálculo da medida Rand *Index*, pois esta calcula a percentagem de decisões corretas e, por conseguinte, se o algoritmo estiver a fundir dois *clusters* num só, teremos nesse *cluster* um determinado número de decisões incorretas que nesta perspetiva deviam ser corretas.

Tabela 61: Tabela de *Ranks* obtida pelo SPSS para o Teste de Wilcoxon – Rand Index- dados de treino.

Ranks

		N	Mean Rank	Sum of Ranks
RISkmeans - RIKC	Negative Ranks	3[a]	3,67	11,00
	Positive Ranks	7[b]	6,29	44,00
	Ties	0[c]		
	Total	10		

a. RISkmeans < RIKC

b. RISkmeans > RIKC

c. RISkmeans = RIKC

Tabela 62: Teste estatístico obtido pelo SPSS para o Teste de Wilcoxon – *Rand Index* – dados de treino.

Test Statistics[a]

	RISkmeans - RIKC
Z	-1,682[b]
Asymp. Sig. (2-tailed)	,093

a. Wilcoxon Signed Ranks Test

b. Based on negative ranks.

e. *Purity* – Resultados dos dados de treino

Finalmente, considerando os resultados da medida *Purity*, constatamos que foi o algoritmo k-C que obteve os melhores resultados para 9 dos 10 repositórios (Tabela 63). Isto significa que foi o algoritmo k-C que agrupou maior percentagem de documentos da mesma forma que a organização manual em 9 dos repositórios.

Observando agora a Tabela 64 podemos concluir que a hipótese nula deve ser rejeitada e portanto os dois algoritmos apresentam resultados diferentes, sendo essas diferenças estatisticamente significativas porque $p = 0{,}009 < 0{,}05$.

Tabela 63: Tabela de *Ranks* obtida pelo SPSS para o Teste de Wilcoxon – *Purity* - dados de treino.

Ranks

		N	Mean Rank	Sum of Ranks
PuritySkmeans - PurityKC	Negative Ranks	9[a]	5,89	53,00
	Positive Ranks	1[b]	2,00	2,00
	Ties	0[c]		
	Total	10		

a. PuritySkmeans < PurityKC

b. PuritySkmeans > PurityKC

c. PuritySkmeans = PurityKC

Tabela 64: Teste estatístico obtido pelo SPSS para o Teste de Wilcoxon – *Purity* – dados de treino.

Test Statistics[a]

	PuritySkmeans - PurityKC
Z	-2,601[b]
Asymp. Sig. (2-tailed)	,009

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

4.8.4. Avaliação da Estabilidade dos Algoritmos

Para avaliar a estabilidade dos algoritmos *Spherical k-means* e k-C, procedemos à construção da Tabela 65 e da Tabela 66, onde se apresentam as diferenças absolutas entre os resultados dos dados do teste e dos dados do treino para cada uma das medidas de avaliação externa utilizadas.

Constata-se que é o algoritmos k-C que apresenta em média as menores diferenças entre os resultados do treino e do teste, para cada uma das medidas de avaliação externa.

Para concluir se as diferenças observadas são significativamente diferentes do ponto de vista estatístico, procedemos de seguida à construção do seguinte teste de hipóteses, que será aplicado a cada uma das medidas de avaliação externa:

H0: Não há diferenças entre o algoritmo *Spherical k-means* e o algoritmo k-C quando se comparam as diferenças absolutas dos resultados dos dados de teste e dados de treino da medida de avaliação externa.

H1: Há diferenças entre o algoritmo *Spherical k-means* e o algoritmo k-C quando se comparam as diferenças absolutas dos resultados dos dados de teste e dados de treino da medida de avaliação externa.

Tabela 65: Resultados das diferenças absolutas entre os resultados do teste e do treino no algoritmo *Spherical k-means*.

Repositórios De Treino	**Spherical k-means**				
	F1	***Precision***	***Recall***	***Rand Index***	***Purity***
D_1	0,288014	0,321305	0,260562	0,105911	0,210233
D_2	0,243899	0,283998	0,213239	0,084912	0,133547
D_3	0,398473	0,370582	0,417394	0,149169	0,301044
D_4	0,279153	0,296633	0,262965	0,104989	0,245963
D_5	0,254017	0,259187	0,24606	0,091745	0,242165
D_6	0,227636	0,266989	0,197573	0,091549	0,188983
D_7	0,182451	0,203765	0,165148	0,070411	0,158594
D_8	0,357474	0,389394	0,330372	0,144198	0,232668
D_9	0,267212	0,275532	0,257113	0,098288	0,271604
D_{10}	0,286762	0,343969	0,242604	0,123498	0,275404
Média	**0,278509**	**0,301135**	**0,259303**	**0,106467**	**0,226021**

Tabela 66: Resultados das diferenças absolutas entre os resultados do teste e do treino no algoritmo k-C.

Repositórios de Treino	**k-C**				
	F1	***Precision***	***Recall***	***Rand Index***	***Purity***
D_1	0,005777	0,036392	0,047018	0,034378	0,024692
D_2	0,101149	0,109599	0,069148	0,0895	0,021487
D_3	0,093731	0,118063	0,041797	0,060875	0,040836
D_4	0,253957	0,188213	0,344239	0,095944	0,201329
D_5	0,172908	0,236995	0,010429	0,152002	0,015195
D_6	0,317501	0,266577	0,386776	0,142355	0,240266
D_7	0,164867	0,133056	0,208837	0,0708	0,150997
D_8	0,197028	0,153153	0,253174	0,080431	0,154796
D_9	0,397965	0,314458	0,499489	0,148462	0,369421
D_{10}	0,22464	0,176688	0,28896	0,093981	0,132004
Média	**0,192952**	**0,173319**	**0,214987**	**0,096873**	**0,135102**

a. F1 – Estabilidade dos algoritmos

Relativamente à medida F1, observa-se na Tabela 67, que o algoritmo k-C é o que apresenta menor diferença absoluta entre os resultados dos dados de treino e de teste para 8 dos repositórios.

Contudo, analisando a Tabela 68 concluímos que não há razões para rejeitar a hipótese nula pois $p = 0{,}093 > 0{,}05$.

Tabela 67: Tabela de *Ranks* obtida pelo SPSS para o Teste de Wilcoxon – F1 – Estabilidade dos algoritmos.

Ranks

		N	Mean Rank	Sum of Ranks
F1KC - F1Skmeans	Negative Ranks	8[a]	5,50	44,00
	Positive Ranks	2[b]	5,50	11,00
	Ties	0[c]		
	Total	10		

a. F1KC < F1Skmeans

b. F1KC > F1Skmeans

c. F1KC = F1Skmeans

Tabela 68: Teste estatístico obtido pelo SPSS para o Teste de Wilcoxon – F1 – Estabilidade dos algoritmos.

Test Statistics[a]

	F1KC - F1Skmeans
Z	-1,682[b]
Asymp. Sig. (2-tailed)	,093

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

b. *Precision* – Estabilidade dos algoritmos

Verifica-se através da Tabela 69, que o algoritmo k-C é o mais estável, porque em 9 dos 10 repositórios apresenta a menor diferença entre os resultados dos dados de teste e de treino. Verifica-se ainda que nesta perspetiva há razões para rejeitar a hipótese nula e aceitar a hipótese alternativa de que os dois algoritmos produzem resultados significativamente diferentes, pois pela observação da Tabela 70, $0{,}013 < 0{,}05$.

Tabela 69: Tabela de *Ranks* obtida pelo SPSS para o Teste de Wilcoxon – *Precision* – Estabilidade dos algoritmos.

Ranks

		N	Mean Rank	Sum of Ranks
PrecisionKC - PrecisionSkmeans	Negative Ranks	9[a]	5,78	52,00
	Positive Ranks	1[b]	3,00	3,00
	Ties	0[c]		
	Total	10		

a. PrecisionKC < PrecisionSkmeans

b. PrecisionKC > PrecisionSkmeans

c. PrecisionKC = PrecisionSkmeans

Tabela 70: Teste estatístico obtido pelo SPSS para o Teste de Wilcoxon – *Precision* – Estabilidade dos algoritmos.

Test Statistics[a]

	PrecisionKC - PrecisionSkmeans
Z	-2,497[b]
Asymp. Sig. (2-tailed)	,013

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

c. *Recall* – Estabilidade dos algoritmos

Tabela 71: Tabela de *Ranks* obtida pelo SPSS para o Teste de Wilcoxon – *Recall* – Estabilidade dos algoritmos.

Ranks

		N	Mean Rank	Sum of Ranks
RecallKC - RecallSkmeans	Negative Ranks	5[a]	6,60	33,00
	Positive Ranks	5[b]	4,40	22,00
	Ties	0[c]		
	Total	10		

a. RecallKC < RecallSkmeans

b. RecallKC > RecallSkmeans

c. RecallKC = RecallSkmeans

Em relação à medida *Recall* não se verificam alterações significativas entre os resultados obtidos pelos dois algoritmos. Na Tabela 71 observa-se que em metade dos repositórios as menores diferenças ocorreram no algoritmo k-C e na outra metade no algoritmo

Spherical k-means. Segundo os resultados obtidos na Tabela 72 concluímos que não há motivos para rejeitar H0, isto porque $p = 0{,}555 > 0{,}05$.

Tabela 72: Teste estatístico obtido pelo SPSS para o Teste de Wilcoxon – *Recall* – Estabilidade dos algoritmos.

Test Statistics[a]

	RecallKC - RecallSkmeans
Z	-,561[b]
Asymp. Sig. (2-tailed)	,575

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

d. *Rand Index* – Estabilidade dos algoritmos

No que ao *Rand Index* diz respeito, verifica-se através da Tabela 73 e da Tabela 74 que as conclusões são as mesmas a que chegamos para a medida *Recall*. Isto significa que também não há razões para rejeitar a hipótese nula.

Tabela 73: Tabela de Ranks obtida pelo SPSS para o Teste de Wilcoxon – *Rand Index* – Estabilidade dos algoritmos.

Ranks

		N	Mean Rank	Sum of Ranks
RIKC - RISkmeans	Negative Ranks	5[a]	6,80	34,00
	Positive Ranks	5[b]	4,20	21,00
	Ties	0[c]		
	Total	10		

a. RIKC < RISkmeans

b. RIKC > RISkmeans

c. RIKC = RISkmeans

Tabela 74: Teste estatístico obtido pelo SPSS para o Teste de Wilcoxon – *Rand Index* – Estabilidade dos algoritmos.

Test Statistics[a]

	RIKC - RISkmeans
Z	-,663[b]
Asymp. Sig. (2-tailed)	,508

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

e. *Purity* – Estabilidade dos algoritmos

Finalmente, segundo a medida *Purity*, verifica-se que o algoritmo k-C é o mais estável em 8 dos 10 repositórios (Tabela 75). Consultando a Tabela 76, constata-se que os dois algoritmos são significativamente diferentes no que respeita à sua estabilidade, uma vez que $p = 0{,}047 < 0{,}05$, permitindo rejeitar a H0.

Tabela 75: Tabela de *Ranks* obtida pelo SPSS para o Teste de Wilcoxon – *Purity* – Estabilidade dos algoritmos.

Ranks

		N	Mean Rank	Sum of Ranks
PurityKC - PuritySkmeans	Negative Ranks	8[a]	5,88	47,00
	Positive Ranks	2[b]	4,00	8,00
	Ties	0[c]		
	Total	10		

a. PurityKC < PuritySkmeans

b. PurityKC > PuritySkmeans

c. PurityKC = PuritySkmeans

Tabela 76: Teste estatístico obtido pelo SPSS para o Teste de Wilcoxon – *Purity* – Estabilidade dos algoritmos.

Test Statistics[a]

	PurityKC - PuritySkmeans
Z	-1,988[b]
Asymp. Sig. (2-tailed)	,047

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

4.8.5. Síntese dos Resultados Obtidos e Propostas de Alteração

Em síntese podemos concluir que os resultados obtidos pelos dois algoritmos nos repositórios de maior dimensão (repositórios de treino) são melhores do que os obtidos nos repositórios de menor dimensão (repositórios de teste).

Nos repositórios de teste as medidas F1, *Recall* e *Purity* indicam que o algoritmo k-C apresenta resultados significativamente melhores do que o algoritmo *Spherical k-means*. Relativamente às medidas *Precision* e *Rand Index*, os resultados não permitem concluir que os algoritmos sejam diferentes tendo em conta estas duas perspetivas.

Relativamente aos repositórios de treino, verificamos que os resultados obtidos nas

medidas *Recall* e *Purity*, indicam que é o algoritmo k-C que apresenta resultados significativamente melhores. Contudo, os resultados da medida *Precision* revelam que o algoritmo *Spherical k-means* obtém resultados significativamente melhores do que o algoritmo k-C. Segundo a medida *Rand Index* não há razões que permitam detetar diferenças significativas entre os algoritmos.

Comparando os algoritmos quanto à sua estabilidade, se utilizarmos apenas como fator de decisão a média das diferenças entre os resultados obtidos nos dados de teste e nos dados de treino, conclui-se que é o algoritmo k-C que é mais estável qualquer que seja a medida utilizada. Contudo, aplicando um teste de hipóteses, apenas as medidas *Precision* e Purity permitem identificar essa diferença entre os algoritmos, sendo que nos restantes casos não há razões para rejeitar a hipótese nula.

Em suma, apenas nos dados de treino usando a medida Precision é que o algoritmo *Spherical k-means* obteve melhores resultados do que o algoritmo k-C mas é necessário relembrar que o algoritmo k-C está a utilizar nestes conjuntos de dados 4 dos 6 grupos esperados pela organização manual pelo que, à partida, já há um conjunto de decisões que serão consideradas erradas, as FP (False Positives).

O facto do algoritmo k-C ter sido executado com 4 *clusters* em vez dos 6 da organização manual, permitiu uma reflexão sobre a necessidade de apresentar mais possibilidades para a escolha inicial das sementes dependendo do contexto. De facto o algoritmo k-C já dispõe de um passo no algoritmo de permite acrescentar novas sementes às detetadas pelo algoritmo de deteção de comunidades. Contudo, se já parte do utilizador a seleção de n tags, no sentido de organizar n grupos de documentos, propomos que seja possível escolher o documento que melhor representa cada *tag*, permitindo escolher numa comunidade mais do que uma semente.

Por outro lado, para além de se permitir ao utilizador a seleção do grau de consenso através da escolha do parâmetro gc (por exemplo se $gc = 3$ significa que só são considerados os documentos em que 3 ou mais utilizadores marcaram aqueles documentos com aquela *tag*), também é, na nossa perspetiva, necessário que o algoritmo de deteção de comunidades tenha em consideração o número de utilizadores que marcou cada *tag* de um documento, pelo que consideramos importante como trabalho futuro modelar um algoritmo que tenha em conta este aspeto.

4.9. Eficiência

Quando analisamos numa perspetiva experimental a complexidade temporal dos algoritmos e a sua consequente escalabilidade, os resultados estão muito dependentes da linguagem de programação e da máquina utilizada. Ainda assim, comparando os tempos médios de execução dos algoritmos *Spherical k-means* e k-C implementados no R utilizando uma máquina cujas características se apresentam na Figura 89, constatamos que os tempos médios de execução nos 10 repositórios de teste e treino são menores no algoritmo *Spherical k-means*. Assim, verifica-se que nos repositórios com 117 documentos o algoritmo *Spherical k-means* é 2,8 vezes mais rápido que o algoritmo k-C. Quando aumentamos a dimensão do repositório para 1053 observa-se que o algoritmo *Spherical k-means* é aproximadamente 13 vezes mais rápido que o algoritmo k-C (Tabela 77).

Estes resultados experimentais indicam que o algoritmo *Spherical k-means* é mais eficiente, o que não constitui surpresa uma vez que já tínhamos concluído que a complexidade temporal do algoritmo k-C sendo $O(n^2)$ é superior à do algoritmo k-means que é $O(kn)$.

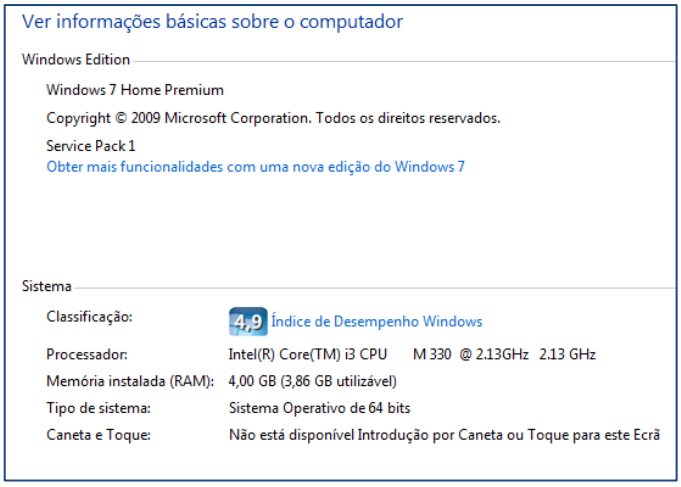


Figura 89: Informações básicas sobre o computador onde foram executados os testes.

Tabela 77: Tempo médio de execução em 10 repositórios com 117 documentos e 1053 documentos.

n	*Spherical k-means*	k-C	$\frac{tempo\ k-C}{tempo\ Sk-means}$
117	28 segundos	76 segundos	2,8
1053	37 minutos	480 minutos	13

Capítulo 5
Conclusões

5.1. Resumo do Trabalho

Com a realização deste estudo procurou-se compreender e que forma o *Tagging* Social pode contribuir para melhorar a eficácia do *Clustering* de documentos.

Assim, partimos para esta investigação com a identificação dos vários tipos de algoritmos de *clustering*, descrevendo os que consideramos mais relevantes. Selecionámos o algoritmo *k-means* como ponto de partida para o nosso trabalho, quer por se ajustar ao *clustering* de texto quer por ser reconhecidamente um algoritmo simples e eficiente, oferecendo grandes potencialidades no que diz respeito à melhoria da sua eficácia pois esta está muito dependente da escolha inicial das sementes e também do tipo de dados.

De forma a sustentarmos as bases do nosso estudo sobre a forma como poderíamos integrar o *tagging* social no algoritmo de *clustering*, começámos por identificar de que modo a natureza das *tags*, vista à luz da Teoria Semiótica segundo Huang e Chuang (2009), poderia contribuir, por um lado para criar as condições para compreendermos que *tags* têm efetivamente impacto no agrupamento dos documentos dependendo do interpretante (ou seja, se é o utilizador de uma comunidade ou se é o autor das *tags*), e por outro para (re)construir o design desta investigação, uma vez que permitiu uma seleção adequada dos repositórios a serem analisados tendo em conta os dois interpretantes indicados.

A primeira opção passou por integrar diretamente as *tags* no VSM em função da sua ocorrência no texto. Assim, as *tags* que não apareciam no texto, as *tags* que apareciam uma vez e as *tags* que apareciam mais do que uma vez eram pesadas de forma diferente

através do parâmetro *Social Slider* (SS). Para esta implementação foi selecionada a similaridade dos cossenos uma vez que após a construção de um modelo teórico de previsão do impacto da integração das *tags* se constatou que aumentando a frequência das *tags* comuns, resultava que os documentos ficavam mais próximos. Ao mesmo tempo que ficavam mais distantes se não partilhassem *tags*.

A segunda abordagem à integração das *tags* consistiu na análise de uma rede de *tags* para determinar especificamente quais as sementes a escolher. Esta abordagem originou um novo algoritmo de *clustering* baseado no algoritmo *k-means*, a que chamámos k-*Communities* (k-C), que inicia tal como o algoritmo *k-means* com k sementes, mas coincidentes com os vetores dos documentos, obtidos através da deteção de comunidades numa rede de *tags*. Quando o algoritmo *k-means* é implementado, o novo centróide é calculado através da média aritmética das coordenadas, estando por isso a ser usadas duas medidas de similaridade, podendo originar centróides incoerentes com a similaridade dos cossenos. Na literatura, a solução que é proposta é normalizar todos os vetores e desta forma passa a interessar mais a sua direção do que a sua magnitude, surgindo assim o algoritmo *Spherical k-means*.

Diferentemente, propomos que o novo centróide seja o documento que está mais próximo dos restantes documentos em cada *cluster*. A desvantagem é que para encontrar este novo centróide é necessário calcular a similaridade dos cossenos entre todos dos documentos que fazem parte de cada *cluster* em cada iteração, tornando-o menos eficiente por iteração do que o algoritmo *k-means*. A análise de complexidade mostra que no pior caso é de $O(n^2)$ por iteração.

Procedemos de seguida à avaliação do resultado do *clustering* para medir a qualidade de um algoritmo, onde foram identificadas duas técnicas principais: o critério externo e o critério interno.

No critério externo são utilizadas medidas que permitem medir o grau de coincidência entre os grupos formados automaticamente, por confronto com os grupos manuais. Contudo, nem sempre é possível conhecer a estrutura do repositório, ou a quantidade de dados pode ser demasiado grande para que seja possível obter uma organização manual dos mesmos. Neste sentido propusemos um algoritmo baseado na informação proveniente das tags e da distância entre os documentos, chamado "*Ground Truth* Automática".

Já no critério interno, as medidas são baseadas na similaridade entre os documentos que fazem parte de um *cluster*, bem como nas diferenças entre documentos colocados em *clusters* diferentes. Neste contexto, e uma vez que estamos interessados em medir a compacidade dos *clusters* usando o critério - quantas vezes a média das distância de cada documento ao seu documento mais próximo dentro de um *cluster* se distancia do seu *cluster* mais próximo? - propusemos um novo índice para medir a avaliação interna, chamado MCI.

A condução dos testes teve em conta a seleção de repositórios em que o interpretante é o autor das *tags* e repositórios cujo interpretante é a comunidade de utilizadores. Os resultados foram criticamente analisados sendo que recorremos a testes estatísticos sempre que o número de repositórios presentes num teste assim o permitia.

5.2. Revisitar os Objetivos da Tese

Na presente secção revisitamos os objetivos da tese, apresentando os resultados obtidos.

5.2.1. Estudar e Analisar Algoritmos para Realizar o *Clustering* de Documentos de Forma Eficiente e Escalável

No Capítulo 1 fazemos um levantamento dos vários tipos de algoritmos de *clustering*, descrevendo detalhadamente o seu princípio de funcionamento, apontando as suas potencialidades e limitações. Da análise dos vários algoritmos, selecionámos o algoritmo *k-means* essencialmente porque é muito eficiente e considerámos que os seus principais problemas, nomeadamente o número e a escolha das sementes iniciais, podiam ser colmatados com a integração do *tagging* social.

5.2.2. Estudar Processos para Redução de Dimensão Espacial e para Pré-Tratamento de Dados

Para além da análise dos vários algoritmos de *clustering*, no Capítulo 1, também apresentamos métodos para representar documentos de texto bem como processos para reduzir a dimensão espacial dos dados, como exemplo temos a técnica LSI e a remoção de palavras que não têm importância semântica, as "*stop words*".

5.2.3. Estudar e Criar uma Possível Integração de uma Classificação Social, com o Agrupamento Automático de Documentos, Baseada no *Tagging* Social

Abaixo apresentamos os principais resultados que conduziram à integração do *tagging* Social no *clustering* de texto.

a. *Tagging* Social

O estudo da integração da classificação social no agrupamento automático de texto teve início com a análise na natureza das *tags* tendo em conta o enquadramento feito por Huang e Chuang (2009) no contexto da Teoria Semiótica. Esta análise permitiu uma clarificação sobre a utilização das tags tendo em atenção o seu interpretante (comunidade de utilizadores, autor das tags e designer do sistema), clarificando a forma como os testes deviam ser conduzidos, não interessando conduzir os testes em função do número de documentos mas sim em função de quem é o interpretante das tags.

No sentido de compreender as características das *tags* atribuídas num sistema procedemos à análise de 5000 documentos do repositório *Wikipedia* tendo-se verificado que em média 57% das *tags* de cada documento foram atribuídas apenas por um utilizador. *Tags* atribuídas por dois utilizadores correspondem a aproximadamente 10%, sendo que esta percentagem tende para 0% à medida que o número de utilizadores que atribui uma mesma *tag* aumenta.

Constatámos ainda que, em média, 15% das *tags* que tinham sido atribuídas por apenas um utilizador já estão representadas por *tags* atribuídas por mais do que um utilizador (tendo por base uma amostra de 25 documentos).

Para além do referido, constatamos que entre as *tags* atribuídas a cada recurso, temos *tags* mais específicas (signo(6)) e *tags* mais gerais como é o caso das *tags* que surgem no signo (8).

Esta análise permitiu tomar algumas decisões, sobretudo quando o interpretante é o utilizador de uma comunidade. Na perspetiva deste interpretante, procura-se o consenso da comunidade no sentido de ver refletidos os interesses da própria comunidade. Deste modo, consideramos que a utilização de todas as *tags* não é necessária para refletir o consenso da comunidade. Sugerimos portanto, agora na perspetiva do designer do sistema, que possa ser dado ao utilizador a possibilidade de escolher o valor do parâmetro gc (grau de consenso) o que permite escolher o número mínimo de ocorrências de uma *tag* para que esta possa ser utilizada para organizar os documentos em grupos que partilham as mesmas *tags*.

Note-se que, esta é uma decisão muito importante sobretudo quando estamos a falar da deteção de comunidades através de uma rede de *tags*. Quanto maior for o número de *tags* diferentes maior será o tempo que o algoritmo demorará a executar.

b. Métodos para Integrar o *Tagging* Social

Foram construídos dois métodos para integrar as *tags*:

- Integração das *tags* diretamente no VSM em função da sua ocorrência no texto. Esta proposta partiu da construção de um modelo teórico de previsão que indica que a utilização da similaridade dos cossenos aproxima documentos que partilham *tags* enquanto afasta os que não partilham.
- Algoritmo k-C, que permite que as sementes sejam escolhidas através de uma rede de *tags*. Para além disso, foi feita uma reflexão critica sobre a utilização das medidas de similaridade utilizadas, conduzindo à alteração da forma como são determinados os centróides no algoritmo *k-means*.

5.2.4. Verificar a Eficácia e Escalabilidade da Integração do *Tagging* Social no Agrupamento Automático de Texto

De seguida, apresentamos a síntese dos principais resultados obtidos em cada um dos testes efetuados.

a. Caso de Estudo I – Repositório da Universidade do Porto

Neste repositório constituído por 142 artigos científicos (particionado em 3 repositórios aproximadamente com a mesma dimensão) começámos por averiguar o impacto da integração das *tags* através do parâmetro SS na eficácia dos agrupamentos. Da análise dos dados verificámos que geralmente foram obtidos melhores resultados quando eram integradas as tags. Contudo, esta melhoria não era proporcional à escolha do parâmetro SS. Por vezes a escolha de SS=5 produzia os melhores resultados, enquanto outras vezes era o parâmetro SS=30, ou seja, o fator social não era preponderante, ou transversal aos testes.

Num outro teste, os mesmos conjuntos de dados foram utilizados mas apenas usando o resumo dos artigos, observando-se resultados piores do que quando era utilizado o texto integral, indicando que menos texto tende a produzir piores *clusters* quando se utiliza o algoritmo *k-means* com se sem integração das *tags*.

Seguiu-se a comparação do algoritmo k-C com o algoritmo *k-means*++ e verificou-se que em média os *clusters* gerados eram mais próximos tanto das classes manuais como das classes geradas através do algoritmo da "*Ground Truth* Automática". Contudo, contrariamente ao observado anteriormente usando o algoritmo k-C, os resultados dos resumos dos artigos são muito similares aos obtidos pelo texto integral. O que parece

indicar que o algoritmo k-C não é influenciado pela utilização de menos texto, sugerindo que é mais estável que o algoritmo *k-means*.

Ainda neste conjunto de dados, comparámos o impacto da integração das *tags* no VSM através do parâmetro SS, usando o algoritmo k-C. Analisando os resultados concluímos que a eficácia dos *clusters* formados não é proporcional ao aumento do parâmetro SS, tal como já tinha sido referido anteriormente. Verificámos ainda que o maior impacto acontece no algoritmo *k-means*++ enquanto que no algoritmo k-C não se observam alterações que justifiquem a integração das *tags* no VSM. Portanto, a decisão foi de passar a utilizar apenas o algoritmo k-C para a realização dos testes posteriores.

Do ponto de vista da análise do algoritmo da "*Ground Truth* Automática", este repositório foi utilizado para comparar com os resultados obtidos quando foram usadas classes manuais, comprovando-se uma grande correlação entre eles.

Finalmente, o cálculo da medida de avaliação interna MCI indica que é o algoritmo k-C o que obtém melhores resultados em 8 dos 9 testes realizados. Verifica-se ainda que, à medida que o parâmetro SS aumenta, também aumenta a média da distância ao *cluster* mais próximo, em comparação com a média das distâncias observadas ao documento mais próximo dentro de cada *cluster*. Este resultado, apesar de confirmar que a utilização da similaridade dos cossenos aproxima os documentos que partilham tags tal como previsto no modelo de previsão, não refletem uma correspondente melhoria da eficácia dos agrupamentos quando aumentamos o parâmetro Social *Slider*.

b. Caso de Estudo II – Repositório de notícias I

Este repositório é constituído por 124 clips de notícias tendo todos os clips sido recolhidos por um único utilizador e todas as *tags* atribuídas por si. Assim, este é um repositório no qual as *tags* são interpretadas na perspetiva do seu autor.

Neste repositório, comparamos o nosso algoritmo k-C com um outro algoritmo de integração das *tags* proposto por Cravino *et al.* (2012), no qual é proposta uma medida de similaridade pesada baseada na similaridade dos cossenos e que tem em atenção uma rede de *tags*. Os resultados obtidos indicam que o algoritmo k-C foi o que obteve melhores resultados, quer usando as classes manuais quer usando as classes automáticas.

c. Caso de Estudo III – Repositório de notícias II

O terceiro repositório é também constituído por clips de notícias colecionadas igualmente por um único utilizador. Neste caso, a comparação foi feita entre o algoritmo k-C e o algoritmo *Spherical k-means*, uma vez que tanto o algoritmo *k-means* como o algoritmo *k-means*++ não funcionaram corretamente neste repositório (gera *clusters* muito desequilibrados, tendo quase todos os clips de notícias sido colocados num único *cluster* enquanto que os restantes ficaram com apenas com 1 clip de notícia cada um).

Neste caso utilizámos como técnica de avaliação o critério relativo, que consiste na alteração de um ou mais parâmetros, no sentido de perceber qual dos algoritmos apresenta melhores resultados.

No algoritmo k-C alterámos o parâmetro que permite incluir novos centróides às sementes iniciais, tendo o k variado entre 7 e 12. O algoritmo *Spherical k-means* foi executado 5 vezes para k=13 (porque era o número de *clusters* manuais); k=10 e k=12 (porque no algoritmo k-C foram os valores de k que obtiveram melhores resultados). A média dos resultados obtidos no algoritmo *Spherical k-means*, nas 5 execuções para cada valor de k selecionado, foi em média inferior aos resultados obtidos pelo algoritmo k-C para os mesmos valores de k.

d. Caso de estudo IV – Repositório *Wikipedia*

Quando temos um repositório de grandes dimensões é necessário delinear muito bem como se vai proceder para realizar o teste pois a utilização de todos os documentos disponíveis forçará uma organização pouco natural dos dados. Neste sentido, foi necessário assegurar que os *clusters* formados faziam ou não sentido pelo que selecionámos *tags* de interesse. Recolhidos esses documentos, analisámos um-a-um se podiam fazer parte de uma classe manual à qual demos o mesmo nome da *tag*. Daqui resultou num repositório com 1170 documentos tendo este repositório sido particionado em 10 repositórios, cada uma com 117 documentos. Testámos os algoritmos *Spherical k-means* e k-C em cada uma destes repositórios e posteriormente formámos mais 10 repositórios, cada uma com 1053 documentos, a que chamámos “dados de treino”.

Os resultados dos dados de teste mostraram que o algoritmo k-C apresenta os melhores resultados, estatisticamente significativos para as medidas F1, *Recall* e *Purity*. Já nos resultados obtidos nos dados de treino, esta diferença significativa só se verifica para as medidas *Recall* e *Purity*. Contudo, é importante referir que o algoritmo k-C está a utilizar k=4, enquanto que as classes manuais são 6. Isto acontece porque o algoritmo de

deteção de comunidades detetou apenas 4 comunidades e o critério para acrescentar novos centroides foi muito restritivo, só permitindo que ao conjunto das sementes fossem acrescentadas novas sementes se a similaridade dos cossenos às restantes sementes fosse 0. Consideramos portanto que é preciso uma reflexão posterior sobre a necessidade de adequar os algoritmos de deteção de comunidades à metodologia que propomos para formar os *clusters*.

Desta forma, na perspetiva do designer do sistema, propomos que seja possível ao utilizador de uma comunidade a seleção de um conjunto de *tags*. No caso de duas ou mais *tags* ficarem na mesma comunidade, devemos identificar nessa comunidade os documentos que estão ligados a mais documentos via respetiva *tag*, alterando assim a forma como o algoritmo k-C seleciona as sementes iniciais (caso o utilizador já tenha pré-definido os seus focos de interesse).

Depois de comparados os algoritmos quanto à sua estabilidade, verificámos que a média das diferenças absolutas entre os dados de teste e de treino, indicam que é o algoritmo k-C que é mais estável qualquer que seja a medida utilizada. Contudo, depois de aplicado um teste de hipóteses, verificámos que apenas para as medidas *Precision* e Purity é possível garantir que essa diferença entre os algoritmos é estatisticamente significativa, sendo que nos restantes casos não há razões para rejeitar a hipótese nula.

e. Análise da Eficiência dos Algoritmos

Efetivamente, o algoritmo k-C é menos eficiente do que o algoritmo k-means. Se utilizarmos repositórios com uma dimensão de 117 documentos, verificámos que o algoritmo *Spherical k-means* é, em média, aproximadamente 2,8 vezes mais rápido que o algoritmo k-C. Por outro lado, quando utilizamos repositórios com 1053 documentos, verificámos que o algoritmo k-C demora, em média, aproximadamente 13 vezes mais a executar do que o algoritmo Spherical k-means.

5.3. Reflexão Crítica sobre o Processo de Investigação

No sentido de responder à questão de investigação – "Um sistema misto de classificação (integração de classificação automática e social) é mais eficaz do que um sistema que integre somente o agrupamento automático de documentos?" – seguimos a seguinte metodologia:

- Revisão de literatura sobre algoritmos de *clustering* e sobre o *tagging* social;

- Com base na revisão de literatura e reflexão sobre as medidas de similaridade utilizadas para implementar os algoritmos, foram propostos dois métodos para integrar as *tags* no algoritmo de *clustering*;
- Para executar os testes utilizámos repositórios em que o interpretante é o autor das *tags* e repositórios em que o interpretante é a comunidade de utilizadores.
- A avaliação dos algoritmos foi feita recorrendo a medidas de avaliação externa e a uma medida de avaliação interna. Para o repositório maior, utilizámos a técnica estatística *cross validation*. Gerámos 10 repositórios de teste e 10 repositórios de treino, tendo os resultados sido analisados recorrendo à formulação de um teste de hipóteses para cada uma das medidas de avaliação externa utilizadas. Para além, disso comparámos os algoritmos quanto à sua estabilidade.

Em resposta à questão de investigação, e considerando os resultados obtidos, podemos afirmar que a utilização do *tagging* social nos repositórios que selecionámos tende a gerar *clusters* mais eficazes. Recordamos, por exemplo, que nos 10 repositórios de teste e nos 10 repositórios de treino, o algoritmo *Spherical k-means* apenas apresentou resultados que do ponto de vista estatístico são significativamente melhores do que o algoritmo k-C quando foi utilizada a medida *Precision* e apenas para os dados de treino, enquanto que o algoritmo k-C apresentou resultados significativamente melhores nos dados de teste para as medidas F1, *Recall* e *Purity*, e nos dados de treino para as medidas *Recall* e *Purity*.

5.3.1. Limitações

Uma das principais limitações encontradas deve-se ao facto da formação de base da autora não ser na área da Informática, o que a limitou em termos de implementação e desenvolvimento dos algoritmos. A execução desta tese decorreu em simultâneo com um projeto internacional de investigação chamado *Breadcrumbs* (ref. UTA-Est/MAI/0007/2009) sendo que, inicialmente, os algoritmos eram implementados por outros investigadores ligados ao projeto, condicionando a liberdade investigativa aquando da exploração de novas opções. Daí decorre que, por exemplo, tanto o algoritmo k-C, como o algoritmo da "*Ground truth* Automática" tenham sido criados analisando os resultados de forma exaustiva (nomeadamente da matriz da distância dos documentos) a partir do Excel e do software NodeXL. Esta limitação traduziu-se em muito tempo despendido, na medida em que os primeiros resultados foram obtidos de forma semiautomática e só posteriormente é que os algoritmos foram implementados.

Resta salientar que não houve tempo para implementar o Índice MCI de forma automática. Este está implementado apenas de forma semiautomática no Excel, tornando o seu cálculo muito demorado quando são utilizados milhares de documentos. Contudo, reconhecemos que seria importante ter estes resultados para completar a nossa análise.

5.4. Contribuição para a Área Científica

Do ponto de vista deste programa doutoral, esta investigação, apesar de incidir com maior preponderância no âmbito da área científica das Ciências da Informação, também incide na área das Ciências e Tecnologias da Comunicação. A primeira porque apresentamos contributos para a organização automática da informação e a segunda devido à utilização do *tagging* social (signo de comunicação online), pois a sua análise permitiu refletir sobre a utilização das *tags* em função do seu interpretante.

5.4.1. Síntese das Contribuições

As contribuições mais relevantes deste trabalho são:

- Análise reflexiva sobre a utilização das *tags* quando a interpretação é feita na perspetiva do autor das *tags* ou na perspetiva da comunidade de utilizadores, baseado no enquadramento da Natureza das tags na perspetiva da Teoria Semiótica apresentado por Huang e Chuang.
- Implementação de algoritmos de *clustering* cuja:
    - integração das *tags* é feita no *Vector Space Model* baseado num modelo teórico criado para prever o impacto da integração das *tags*;
    - rede de *tags* é utilizada para determinar as sementes iniciais;
- No contexto da avaliação dos algoritmos foram criados dois algoritmos
    - Algoritmo da "*Ground Truth* Automática" que permite executar as medidas de avaliação externa quando se desconhece a estrutura do repositório.
    - MCI (*Maximum Cosine Index*) índice que mede a distância entre cada *cluster* e o seu *cluster* mais próximo e determina quantas vezes é superior à média das distâncias entre cada documento e o seu documento mais próximo dentro de cada *cluster*.

5.4.2. Publicações Decorrentes da Investigação

Listamos de seguida as publicações efetuadas no decorrer desta investigação:

Cunha, E., Figueira, Á., & Mealha, Ó. (2013). *Clustering and Classifying Text Documents - a Revisit to Tagging Integration Methods.* Paper presented at the 5th International Conference on Knowledge Discovery and information Retrieval (KDIR 2013) (pp. 160-168), Vila Moura, Portugal.

Cunha, E., Figueira, Á., & Mealha, Ó. (2013). *Clustering Documents Using Tagging Communities and Semantic Proximity.* Paper presented at the 8th Iberian Conference on Information Systems and Technologies (CISTI) (Volume I, pp. 591-596), Lisboa, Portugal.

Cunha, E., Figueira, Á., & Mealha, Ó. (2013). *Eficácia e Eficiência em Agrupamentos Semânticos Automáticos.* Paper presented at the 8th Iberian Conference on Information Systems and Technologies (CISTI) (volume II pp. 315-318), Lisboa, Portugal.

Cunha, E., & Figueira, Á. (2012). *Automatic Clustering Assessment through a Social Tagging System.* Paper presented at the 15th IEEE International Conference on Computational Science and Engineering (pp. 74-81), Paphos, Cyprus.

5.5. Trabalho Futuro

Como referido anteriormente, uma das intenções de trabalho futuro prende-se com a implementação automática do índice de avaliação interna MCI, de forma a que possamos utilizar os resultados daí decorrentes para complementar a investigação desenvolvida.

Para além disso, existe ainda a intenção de desenhar um algoritmo de deteção de comunidades que permita construir as comunidades dando maior importância às tags atribuídas por mais pessoas a um determinado documento.

Finalmente, é importante testar se o algoritmo *Spherical k-means*, complementado com algumas características do algoritmo k-C (nomeadamente, a mesma seleção inicial das sementes e a possibilidade de acrescentar novas sementes), produz resultados semelhantes (que, confirmando-se, justificariam a sua utilização de forma a melhorar a eficiência).

Referências

Arthur, D., & Vassilvitskii, S. (2006). *How Slow is the k-means Method?* Paper presented at the Proceedings of the twenty-second annual symposium on Computational geometry (pp. 144-153), Sedona, AZ, USA.

Arthur, D., & Vassilvitskii, S. (2007). *k-means++: the Advantages of Careful Seeding*. Paper presented at the Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (pp. 1027-1035), New Orleans, Louisiana.

Berry, M., & Castellanos, M. (Eds.). (2008). *Survey of Text Mining II - Clustering, Classification, and Retrieval*: Springer.

Borko, H. (1968). *Information Science: What Is It?* American Documentation, 19(1), 3-5.

Bush, V. (1945). *As We May Think*. Atlantic Monthly, v. 176(1), pp. 101-108.

Carmo, H., & Ferrreira, M. M. (2008). *Metodologia da Investigação - Guia para auto-aprendizagem* (2.ª edição): Universidade Aberta.

Castells, M. (2000). *A Era da Informação: Economia, Sociedade e Cultura*. Volume I. A Sociedade em Rede: Fundação Calouste Gulbenkian Serviço de Educação e Bolsas.

Clauset, A., Newman, M. E. J., & Moore, C. (2004). *Finding community structure in very large networks*. Physical Review E, 70, 066111.

Cravino, N., Devezas, J., & Figueira, Á. (2012). *Using the Overlapping Community Structure of a Network of Tags to Improve Text Clustering*. Paper presented at the 23rd ACM Conference on Hypertext and Social Media (HT 2012) (pp. 239-244), Milwaukee, WI, USA.

Cunha, E., & Figueira, Á. (2012). *Automatic Clustering Assessment through a Social Tagging System*. Paper presented at the 15th IEEE International Conference on Computational Science and Engineering (pp. 74-81), Paphos, Cyprus.

Cunha, E., Figueira, Á., & Mealha, Ó. (2013a). *Clustering and Classifying Text Documents - a Revisit to Tagging Integration Methods*. Paper presented at the 5th International Conference on Knowledge Discovery and information Retrieval (KDIR 2013) (pp. 160-168), Vila Moura, Portugal.

Cunha, E., Figueira, Á., & Mealha, Ó. (2013b). *Clustering Documents Using Tagging Communities and Semantic Proximity*. Paper presented at the 8th Iberian Conference on Information Systems and Technologies (CISTI) (Volume I, pp. 591-596), Lisboa, Portugal.

Davies, D. L., & Bouldin, D. W. (1979). *A Cluster Separation Measure*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-1(2), 224-227.

Deerwester, S., Dumais, S., Furnas, S. T., Landauer, T. K., & Harshman, R. (1990). *Indexing by Latent Semantic Analysis*. Journal of the American Society for Information Science, 41(6), 391-407.

Demsar, J. (2006). *Statistical Comparisons of Classifiers over Multiple Data Sets*. J. Mach. Learn. Res., 7, 1-30.

Dunn, J. C. (1974). *Well Separated Clusters and Optimal Fuzzy-Partitions.* Journal of Cybernetics, Vol. 4 pp. 95-104.

DYE, J. (2006). Folksonomy : *A Game of High-Tech (and high-stakes) tag*. (Vol. 29). Wilton, CT, ETATS-UNIS: Online.

Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*. Paper presented at the 2nd International Conference on Knowledge Discovery and Data Mining (pp. 226-231), Portland, Oregon.

Feldman, R., & Sanger, J. (2007). *The Text Mining Handbook Advanced Approaches in Analyzing Unstructured Data (1.ª ed.)*: Cambridge University Press.

Fortunato, S., & Castellano, C. (2009). Community Structure in Graphs Encyclopedia of Complexity and Systems Science (Vol. Part 3, pp. 1141-1163).

Ghizzi, E. B. (2009). *Introdução à Semiótica de Charles S. Peirce* - texto de apoio didático. 2012

Girvan, M., & Newman, M. E. J. (2002). *Community Structure in Social and Biological Networks*. Paper presented at the Proceedings of the National Academy of Science (Volume 12, pp. 7821–7826).

Grossman, D., & Frieder, O. (2004). *Information Retrieval - Algorithms and Heuristics*: Springer.

Halkidi, M., & Vazirgiannis, M. (2001). *Clustering Validity Assessment: Finding the Optimal Partitioning of a Data Set*. Paper presented at the IEEE International Conference on Data Mining pp. (187 – 194), San Jose, CA.

Huang, A. (2008). *Similarity Measures for Text Document Clustering*. Paper presented at the New Zealand Computer Science Research Student Conference (pp. 49-56), Christchurch, New Zealand.

Huang, A. W., & Chuang, T. (2009). *Social Tagging, Online Communication, and Peircean Semiotics: A Conceptual Framework*: Journal of Information Science, 35, 340-357.

Kao, A., & Poteet, S. (2005). *Text mining and natural language processing: introduction for the special issue*. SIGKDD Explor. Newsl., 7(1), 1-2.

Kaplan, A. M., & Haenlein, M. (2010). *Users of the world, unite! The challenges and opportunities of social media*. Business Horizons 53 (1), p. 61.

Konchady, M. (2006). *Text Mining Application Programming*: Charles River Media.

Lee, C. S., Goh, D. H.-L., Razikin, K., & Chua, A. Y. K. (2009). *Tagging, Sharing and the Influence of Personal Experience*. Journal of Digital Information, 10(1), Available at: http://journals.tdl.org/jodi/article/view/275/275.

Levy, P. (1997). *Collective Intelligence: Mankind's Emerging World in Cyberspace*: Perseus Books.

Lohmann, S. (2011). *Social Tagging and Folksonomies*. Retrieved 2012: http://www.socialtagging.org/.

Lovins, J. B. (1968). *Development of a Stemming Algorithm*. Mechanical Translation and Computational Linguistics 11, 22–31.

MacQueen, J. B. (1967). *Some Methods for Classification and Analysis of MultiVariate*. Paper presented at the Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability (Volume 1, pp. 281-297), Berkeley, California

Manning, C., Raghavan, P., & Schütze, H. (2009). *An Introduction to Information Retrieval*: Cambridge University Press. Cambridge, England.

Maslov, A., Mikeal, A., Weimer, K., & Leggett, J. (2009). *Cooperation or Control? Web 2.0 and the Digital Library*. Journal of Digital Information, 10(1).

Newman, M. E. J. (2004). *Fast algorithm for detecting community structure in networks*. Physical Review, E 69.

Newman, M. E. J., & Girvan, M. (2004). *Finding and Evaluating Community Structure in Networks*. Physical Review E, 69(2), 026113.

O'Reilly, T. (2007). *What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software*. International Journal of Digital Economics, N0. 65, pp. 17-37.

Paige, C. C., & Saunders, M. A. (1981). *Towards a Generalized Singular Value Decomposition*. SIAM Journal on Numerical Analysis, 18(3), 398-405.

Pavan, K., Rao, A., Rao, A. V., & Sridhar, G. R. (2010). *Single Pass Seed Selection Algorithm for k-Means*. Journal of Computer Science 6(1), 60-66.

Peirce, C. S. (1958). Collected papers of Charles Sanders Peirce. Cambridge, MA: Harvard University Press.

Ramage, D., Heymann, P., Manning, C. D., & Garcia-Molina, H. (2009). *Clustering the Tagged Web*. Paper presented at the Proceedings of the Second ACM International Conference on Web Search and Data Mining (pp. 54-63), Barcelona, Spain.

Salton, G., Wong, A., & Yang, C. S. (1975). *A Vector Space Model for Automatic Indexing*. Commun. ACM, 18(11), 613-620.

Sander, J., Ester, M., Kriegel, H.-P., & Xu, X. (1998). *Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications*. Data Min. Knowl. Discov., 2(2), 169-194.

Silva, A. M. d. (2000). *A gestão da informação arquivística e suas repercussões na produção do conhecimento científico*. Paper presented at the Seminário Internacional de Arquivos de Tradição Ibérica.

Silva, A. m. d. (2006). *Documento e Informação : As Questões Ontológica e Epistemológica*. In P. U. d. P. F. d. Letras (Ed.).

Springer, M., Dulabahn, B., Michel, P., Natanson, B., Reser, D., Woodward, D., et al. (2008). For The Common Good: The Library of Congreess. Flichr Pilot Project – Report Summary.

Tasdemir, K., & Merenyi, E. (2011). *A Validity Index for Prototype-Based Clustering of Data Sets With Complex Cluster Structures*. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 41(4), 1039-1053.

Theodoridis, S., & Koutroumbas, K. (2009). *Pattern Recognition, Fourth Edition (Fourth Edition ed.)*: Academic Press.

Trant, J. (2008). Tagging, Folksonomy and Art Museums: Results of steve museum's rechearch. from http://verne.steve.museum/SteveResearchReport2008.pdf

Trant, J. (2009). *Studying Social Tagging and Folksonomy: A Review and Framework*. Journal of Digital Information. Special Issue on Digital Libraries and User Generated Content.

Turing, A. (1950). *Computing Machinery and Intelligence*. Mind 49, 433-460.

Wakita, K., & Tsurumi, T. (2007). *Finding Community Structure in Mega-Scale Social Networks: [extended abstract]*. Paper presented at the Proceedings of the 16th international conference on World Wide Web (pp. 1275-1276), Banff, Alberta, Canada

Wal, V. (2007). Folksonomy Coinage and Definition. http://vanderwal.net/folksonomy.html

Witten, I. H., & Frank, E. (2005). *Data Mining Practical Machine Learning Tools and Techniques*: MORGAN KAUFMANN PUBLISHERS ELSEVIER.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., et al. (2007). *Top 10 algorithms in data mining*. Knowl. Inf. Syst., 14(1), 1-37.

Zhong, S. (2005). *Efficient Online Spherical k-means Clustering*. Paper presented at the Prokhorov, D. (Eds.), Proceeding of the IEEE international joint conference on neural networks (IJCNN 2005) (Volume V, pp. 3180 - 3185), Montreal, QC, Canada.